$24.95

Understanding Device Level Buses

A Tutorial

By Bob Svacina

About the Author:
Bob Svacina

Bob has been in the electrical industry since 1973. For 17 years, he has served in positions in field sales, management, service, and plant start-up. These field assignments have included gas and oil in Texas, grain handling and agri-chem in the Midwest, and adhesives and laminates in the Upper Midwest.

At age 16, Bob started working for the Union Pacific Railroad in the Transportation Department electronically collecting and processing car loading and train movement data. At age 21, during summer vacations and holidays, he was a shift supervisor managing transportation data for the entire Union Pacific network.

Bob was a pioneer in the use of PCs, PLCs™*, and Ethernet™* - Novell™* Networks in the agri-chem industry in the mid 1980's.

Bob joined TURCK in 1991 as a marketing specialist for their automation products. He wrote Understanding Hazardous Area Sensing in 1993. At TURCK, he has managed the early development of field nodes for various device level buses. His past experience and education in Physics and Chemistry have proved to be an ideal background for device level networks and Intrinsic Safety.

Warranties and Disclaimers

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. TURCK INCORPORATED ASSUMES NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THIS PUBLICATION OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS PUBLICATION.

REFERENCES TO CORPORATIONS, THEIR SERVICES AND PRODUCTS, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO EVENT SHALL TURCK INCORPORATED BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS INFORMATION.

THIS PUBLICATION COULD INCLUDE TECHNICAL OR OTHER INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. TURCK INCORPORATED MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Trademarks

TURCK and Sensoplex, are trademarks of TURCK Incorporated or its subsidiaries which may be registered in certain jurisdictions.

*All other brand and product names are trademarks, registered trademarks or service marks of their respective holders.

Table of Contents

Preface

Chapter 1

— The Fifth Stage of the Industrial Revolution —

Chapter 2

— What's on the Bus —

Chapter 3

— Bus Talk —

Chapter 4

— The Bus Tour —

Chapter 5

— Bus Access —

Chapter 6

— A Choice of Device Level Buses —

Chapter 7

— Common Relief for a Sick Bus —

Appendix A

— Error and duplicate address detection when using a Phase Continuous Frequency Shift Key Modulated Signal —

Appendix B

— ISO-OSI Seven Layer Network Model —

Appendix C

Preface

The Buses Are Here

This tutorial is about device level buses. In general, think of them as an industrial telephone line that connects sensors, pushbuttons and actuators to some type of controller, such as a PLC or industrial computer.

Device level buses are not all alike; there is a variety to choose from–at least six general purpose buses and two prominent specialty buses. Each one is distinctive and has features important to someone. Chapter 6, A Choice of Device Level Buses, describes the features and benefits that differentiate these buses. In addition, this tutorial will briefly discuss the supporting groups and the significance of infrastructure.

The question often comes up: "Are these buses ready?" Yes, they are ready to do common industrial tasks that we ask of control systems today. These involve inputs from pushbuttons, sensors, and contacts, and outputs to pilot lights, solenoids, and contactors. Bus features that have promise include reconfiguring drives and instruments over a network to make a new part or blend a new recipe.

Hot replacement of industrial network components is a hot topic today. In the case of your desktop PC, most hardware, software, or other configuration changes require a fresh boot from an idle or off condition. As a result of this, hot replacement of components on a PC is very difficult and the same holds true for an industrial network. In a few years, many things may be possible.

Multiple disciplines

Buses are difficult to dissect and discuss, since several disciplines are involved. Jargon is a problem. We mix old terms from the early days of the telegraph and advanced terminology used in modern computers. Until 1985 when LANs (Local Area Networks) were integrated into many business applications, networks were the parochial domain of a few mainframe computer manufacturers, military communications, and telephone companies. Novell, IBM's ™* AT class PC and Ethernet changed that, making it possible for a person with general technical training and patience to start up a network and maintain it.

For a device level network to succeed today, the training level should not to exceed the training required for the early office networks. In fact, it really should be less demanding to learn how to maintain device level networks. Unfortunately, this is not the case. Early office networks put only three things on a network:

1) file server;
2) work stations; and,
3) a printer.

It would hardly be worthwhile to build an industrial network with only three types of bus occupants. The technical properties change little from one bus occupant to another. But jargon from manufacturer to manufacturer, from discipline to discipline, and among the entire sales and service channels varies widely. This tutorial uses simple language whenever possible. It may be neat to sound like a cyberspace master, but what's the point if only a few other people understand? These networks will cause an industry change in the same magnitude as the PLCs did years ago. Such a fundamental change will affect nearly everyone in the industry. The techie language isn't bad on paper but when a machine is down, we need words everyone understands.

Some of this information may seem "old hat", especially if it pertains to a discipline you already understand. But hopefully other topics will be new and informative. If one or more areas are of

particular interest to you, you may want to look up additional information. The Instrument Society of America is an excellent source of training and text. Also, many colleges and technical schools offer data communications courses. Industrial data communication is not new, but definitely coming of age. I suggest talking with the instructors of these courses. They may be very willing to enhance the portion of the course that applies to factory floor data.

Drums and smoke signals

We really aren't going back that far. But those were forms of serial data, data encoding, and protocol. Drums and smoke signals were an economical way to communicate with people some distance away. And economy continues to be one of the most universal benefits of serial data communication. Today, two wires can communicate with hundreds of sensors and actuators.

Another benefit, one that has become as important as economy, derives from the diagnostics that can be added. Device level buses require intelligence to communicate. And, since some form of microprocessor is already available, a few lines of diagnostic code can be written for almost no cost. Types and locations of problems can be transmitted over the bus. And, in some cases, this can even be done proactively before a problem causes a crash-type shutdown.

The last major benefit that buses offer is flexibility. Many products have marketing life cycles that are shorter than the design and start-up phase. Buses allow for quick changes and most of the components are reusable. Changeovers that took weeks, can be done in days. Connectorized buses make it happen. I am avoiding the term "plug-and-play", the industry is not there yet, except for some simple systems, but it is very close.

Buses are here. And the change is more fundamental than most people will admit out loud. Conveyor manufacturers are looking at buses and saying they should redesign the conveyor from the bottom-up to maximize the benefits of buses. The diagnostics are another benefit that will cause long-term changes. Instead of building a fortress around an entire control system, more users are looking at a plug-and-receptacle control system that uses buses with capable diagnostics. Many of the auto manufacturers have already taken this step. Their goal is to fix absolutely anything in 15 minutes. It just cannot be done with traditional pipe and wire systems.

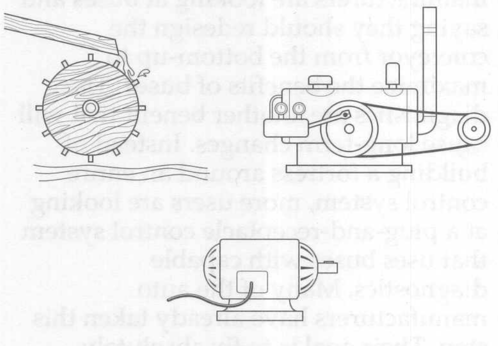
As a final note, please remember that buses are dynamic, which means that some information in this tutorial will soon be outdated.

Chapter 1

The Fifth Stage of the Industrial Revolution

Power and control

The fourth stage of the industrial revolution began in 1974 when Dick Morley introduced what was then called the Programmable Controller. In retrospect, no single part of it was revolutionary; but the programmable controller started the fourth stage of the industrial revolution.



First three stages of the industrial revolution

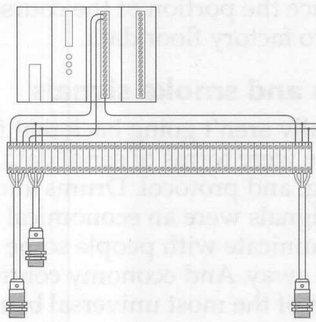
Figure 1 - 1

The previous three stages relied upon water, steam and electricity as the prime sources of power to run tools, but people still controlled the tools. The programmable controller changed things. Now the controller runs the tools and people run the controller.

Interconnect wiring

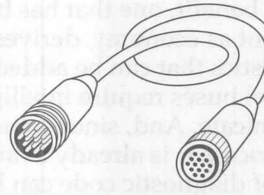
Parallel wiring is used to transfer information from the input devices to the controller and from the controller to the output devices. This is also called point-to-point wiring.

On large jobs, several termination stages are required between the controllers and the field devices. A single automotive assembly line could have as many as 250,000 termination points, each being a potential failure point. To cut down on installation time, special multi-core cables, pre-made off-site, are used. This reduces installation time, but requires long shut-downs to track down and repair a broken wire.



Traditional control system using parallel wiring

Figure 1 - 2



Multi-core cable with male and female connector

Figure 1 - 3

The early buses

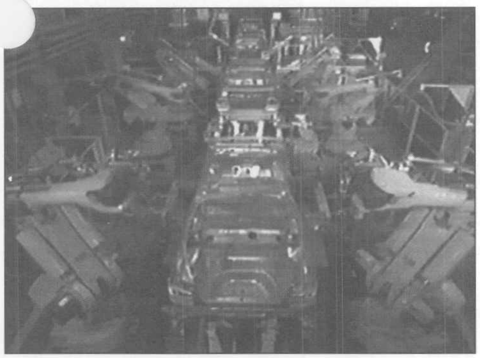
At about the same time the Programmable Controller was introduced, a product called Directrol™* was introduced by Cutler-Hammer™*. This was the first device level fieldbus system. It was ahead of its time. Most of today's buses still can't match some of the features of Directrol. It died quietly in the early 1980s, with only a few dedicated customers remaining in power plants, paper mills, and airport facilities. It was simply too big. The PCs and PLCs of those days did not have the horsepower to match Directrol.

From 1985 to 1987 three device level buses were introduced that are still alive today; General Electric's Genius I/O, Phoenix's Interbus-S, and TURCK's Sensoplex®.

Genius I/O is versatile and can handle large amounts of data. As well as handling I/Os, as its name implies, it can pass information from PLCs to other PLCs and man-machine interface panels.

Interbus-S is clock-work punctual over huge distances. Phoenix has opened this bus to a vendors group. Its speed and dependable scans have made this an early choice of drive manufacturers.

Sensoplex® is rugged and mounts right on the machine in the worst environments. The second generation was introduced in 1992. Sensoplex® is a specialty bus. Both generations were designed to operate in close proximity to the destructive magnetic fields generated around welders. Another variation, introduced in 1991, was approved as Intrinsically Safe; it can operate in all except the most explosive environments such as Acetylene and Hydrogen.



Automotive body welding
Figure 1-4

Hazardous area
Figure 1 - 5

The general purpose "Open" buses

Several other buses have been introduced since 1991. Now most electrical electronic control manufacturers have products or projects for one or more "Open" buses. "Open" may be the most subjective word in this tutorial. Since "O" as in "Open" represents the first letter of the ODVA (Open DeviceNet Vendor Association), let's take a look at "Open" with respect to DeviceNet.

- The CANbus chip that is the basis for DeviceNet is licensed to many Silicon manufacturers–Motorola, Intel, Hitachi, and Phillips, to name a few.
- DeviceNet specifications and updates are available for a nominal fee.
- DeviceNet conformance testing software and hardware is available for a nominal cost. The software and specifications have been reviewed, modified, and tested by engineers from Allen-Bradley, Dearborn Group, Eaton, Omron, SMC, TURCK, and others.
- A vendor-sponsored organization with published and registered bylaws allows different levels of membership, from users to DeviceNet vendors. Annual meetings are open to members.

Specialty buses

TURCK's Sensoplex® is a specialty bus that works around welders and in explosive areas. Maybe other specialty bus will debut as the premier bus around RF furnaces, or another around the huge DC drives on a drilling rig, or yet another that will control a railroad switch yard in Siberia in the dead of winter. Specialty buses solve special problems.

Very few buses are completely open. Often some portion of the bus is closed or limited to a single or small group of vendor(s) usually for performance or marketing reasons.

Performance reasons are technically easy to understand. Consider the maintenance of a Porsche 988 as an analogy. One wouldn't use just any old parts, or a discount tire store's tools to tune and maintain the engine. Maintenance of the 988 should be performed by a closed or limited group of professionals. Likewise, specialty buses designed for performance need to be limited to a closed group of professionals who can implement and maintain performance standards.

Marketing reasons for closed buses are harder to understand. For example, a user might become frustrated because one product cannot be hooked to another product to get the best solution for his business. It is very frustrating if his global competitor in another country can hook the two together.

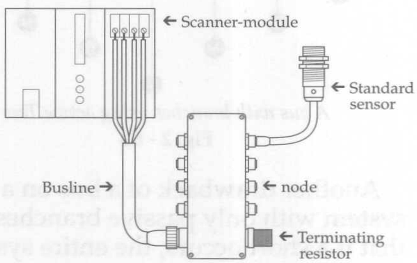
Personally, I believe that performance is the only justification for a closed bus. Users are competing on a worldwide basis. If machinery advances are inhibited because of artificially closed systems, the customer suffers the loss. And eventually the customer can be lost, as stated by Sam Walton: "There is only one boss. The customer. And he can fire everybody in the company from the chairman on down, simply by spending his money somewhere else".

Chapter 2

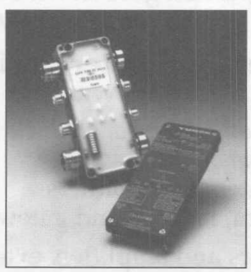
What's on the Bus

ɪ ne talkers, the listeners, and the busline

Many levels of data may be transferred over a bus from an ON-OFF condition to a large graphics file. Buses can be light-duty or rugged. Some can even control most of the functions on a combat aircraft. Instead of talking about all buses and all applications at once, we'll first look at a small bus system (figure 2-1). It has one mini PLC with an interface to the bus and one node that supports inputs and outputs. A *node* (figure 2-2) is an addressable physical point on the bus. The interface is technically called a programmable gateway, but the commonly accepted term is scanner-module or scanner-card. One of the lowest node addresses (usually ZERO or ONE) is reserved for this scanner-module.



PLC and bus system with an Input-Output node
Figure 2-1

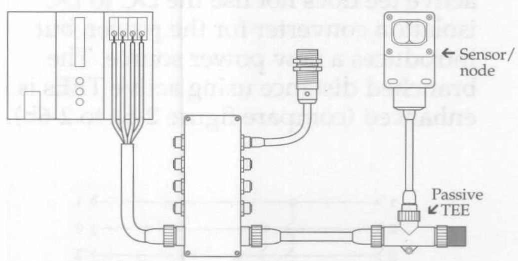


A potted Input-Output node for on-the-machine mounting with DIP switches for manual or network addressing, shown in the network addressing position. *Courtesy TURCK Inc.*
Fig. 2 - 2

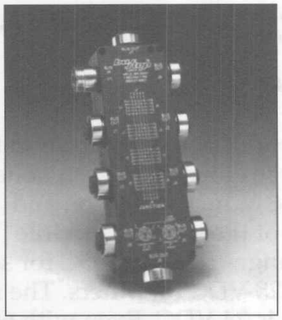
A source of power is required to operate the chipsets in each node, any LEDs, the inductive sensor, and portions of the scanner module. This power ranges from 12 VDC for some buses to 28 VDC for others. The most common is 24 VDC. Even with 60 nodes, one or two power supplies for a bus usually hit the sweet spot for economical operation. Individual power supplies for each node and the associated wiring would make the cost of the system hard to justify. The power and the data are bused from node to node via a *busline*. This busline varies from bus to bus. (More is said about this topic later.) The *data lines* of most buses also require a *terminating resistor*. The terminating resistor balances and tunes the data lines. Again, this specification varies from bus to bus.

Branches

Let's add more to this simple network (see figure 2-3). The next thing we'll add is a *single drop passive hub*; also called a *passive TEE*. It allows the busline to be branched. Of course, if there is a single drop passive hub, then there must be a *multiple drop passive hub*.



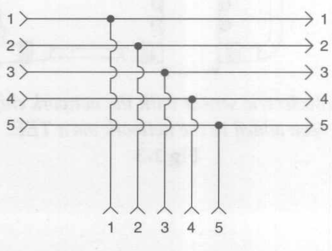
A photoelectric sensor with the network chipset has been added to the network via a TEE.
Fig 2-3



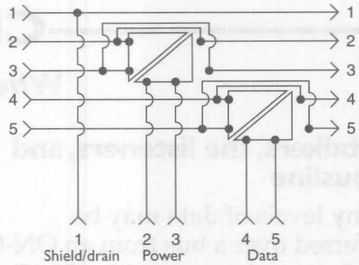
A 6 port passive hub (Courtesy TURCK Inc.)
Fig. 2-4

Many buses have restrictions on these passive branches. This is because of the distortion that occurs when part of a signal going down the main busline goes down the branch. This is cumulative. The longer the branches and the more of them there are, the more the original signal is distorted.

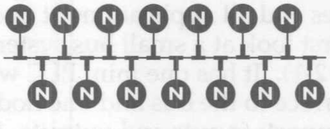
An *active TEE* would offer a solution to this problem. Figure 2-5 compares a passive TEE to a 5-wire active TEE. This example shows a bus with two power lines, two data lines and a fifth wire which serves as the drain. The passive TEE is straightforward. The active TEE isolates the data lines with *bi-directional optical amplifiers,* and the power is isolated with a *DC to DC isolation converter* that has a non-destructive current limit. Another variation of the active tee does not use the DC to DC isolation converter for the power, but introduces a new power source. The branched distance using active TEEs is enhanced (compare figure 2-6a to 2-6b).



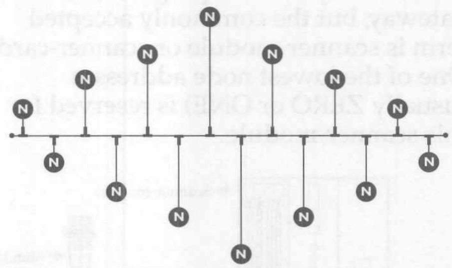
Passive TEE
Figure 2 - 5a



Active TEE
Figure 2 - 5b



A bus with restrictive branches using passive TEEs
Fig. 2 - 6a



A bus with branches using active Tees
Fig. 2 - 6b

Another drawback of a bus on a system with only passive branches is that if a short occurs, the entire system goes down. In contrast, if a short develops in a branch from an active TEE, that portion goes down but the rest of the system remains active.

Repeaters, amplifiers, bridges, routers, and gateways

The two other types of products that may also be found on the bus:

- Repeaters or amplifiers
- Bridges, routers, and gateways

Repeaters and amplifiers extend the distance of a bus. Bridges, routers, and gateways make it possible to link one bus to another. The bridge links segments of similar buses. It is more advanced than a repeater. Repeaters

transfer all messages from segment to segment. Bridges selectively transfer messages from segment to segment. A bridge transfers the message only when the destination is on the linked segment. All other times it keeps the local messages local. There are only two choices for a bridge, send the message to the linked segment or keep it local.

Routers are still more advanced. Their job is to get a message from the source to the destination even when there may be hundreds of choices and hundreds of segments. Their expertise is reading the address and then sending the message on the right route. More often than not the message will go through many routers as it travels from source to destination.

A gateway is more advanced than the router. It can link completely dissimilar buses. These products will be described in more detail in Chapter 4 under the topic "The Bus Routes and Transfers".

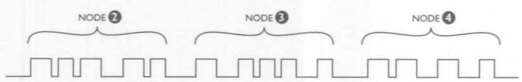
Diagnostic Tools

Diagnostic tools can be as simple as a smart terminating resistor with LEDs that indicates acceptable voltage, and possibly the presence of communication on the data lines. At the high end is a portable computer with an interface card that is connected to the bus. With the right hardware and software, the bus can be given a complete physical. Ill-behaved *nodes* can be detected, unexpected *signals* can be logged, and timing errors found. Signals are discussed in the next chapter under the topic, The Signal.

Multiplexed Serial Data Transfer

Serial data transfer is how information travels on a bus. This is *binary information*. What it does is fairly simple; how it does it is more complex. A long explanation of this topic would require several text books. One good application-oriented text is "RS-232 Simplified", by Byron Putman, Prentice Hall Press, 1987. The short version relies on the reader's blind faith that computers can really talk to each other over two wires.

The serial data transfer involves one sender and at least one or more listeners (receivers). Both the sender and the receiver(s) typically must make a connection and establish some form of mutual time base (broadcast buses and Smart Distributed System do not make a connection). Then, the data is transferred as a string of bits (ON-OFF states).



Mutiplex serial data transfer
Figure 3 - 1

An industrial bus is likely to have several nodes. Most of the device level buses have been designed for either 32, 64, 128, or 256 possible addressable nodes.

The Signal

There are two common types of signals, *modulated and square wave*. All of the buses discussed in Chapter 6 use either one or the other with only one exception, ASI, and it is composite of the two. Modern business couldn't exist without both. One can use the fax to legally bind a real estate agreement (in some states), order a pizza, request a favorite song, or even make a political statement to a favorite talk show host.

Internally, the fax moves data with square wave signals between the photocells that read a document, the CPU, its memory, and the printer. Externally, serial data uses a modulated signal to send data to another fax machine. Why two types of signals? Each one has pros and cons; therefore, a trade-off is involved in deciding which to use.

Square wave signal PROS:

- Economical; there is huge variety of competitively priced components to handle square wave signals.
- More data for a given cable bandwidth with low-cost transmission media and components.
- Possible to compress data.

Square wave signal CONS:

- Becomes distorted as it travels down copper media.
- Must be repeated (not amplified) to extend transmission distance, therefore it has a longer wait state than a modulated signal.

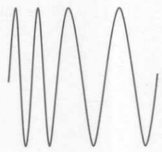
Modulated PROS:

- Less prone to distort on long runs, signal becomes weaker. Therefore, it can be amplified in near real time (see figure 3-2a and 3-2b).
- Potentially more data per unit time with high cost transmission media and components.

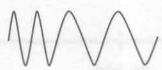
Modulated CONS:

- Components are somewhat more costly than those used for square wave signals.
- Sometimes the circuit boards are larger to accommodate the components needed for a modulated signal. This may result in a larger finished product.

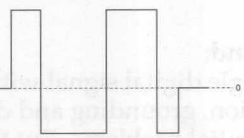
- Must be translated to a square wave signal to be read by CPU or moved into memory.



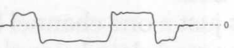
Fresh sine wave modulated signal
Figure 3-2a



Modulated signal near the maximum transmission distance
Fig 3-2b



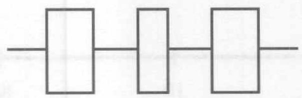
Ground referenced square wave signal. RS-232 or 432
Figure 3-3a



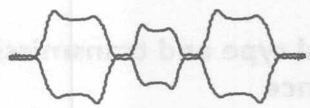
Ground referenced square wave signal near the maximum transmission distance. RS-232 or 432
Figure 3-3b

Inductance and capacitance of the wire distort the high frequency components of the ground referenced square wave signal as it travels down the wire (see figure 3-3a and b). Figure 3-4a and b shows a differential digital square wave signal.

So what is the difference between a differential signal and a ground reference signal? The EIA RS-422 & 485 are differential signals – the conventionally identified 0 voltage doesn't exist–it is derived (see figure 3-4a and b). The EIA RS-232 and 423 are referenced to ground – although the 0 voltage may not be grounded, it is referenced to ground (see figure 3-3a and b). The shorter the distance, the better the ground reference signal is. From chip to chip, on a circuit board, or the few meters from your computer to an external modem or a printer are good applications for the ground reference square wave. But the longer the distance, the more problems we have with referencing either state (high or low) to ground.



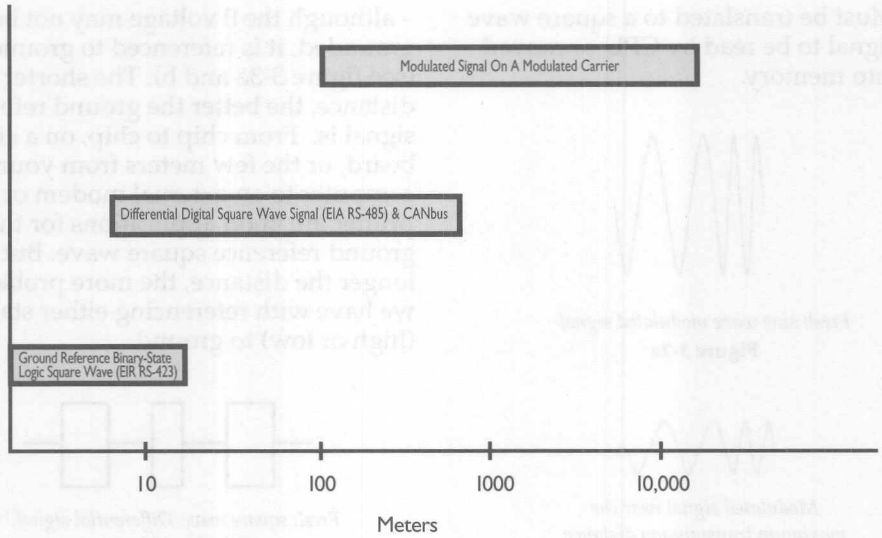
Fresh square wave Differential signal, RS-422, 485, etc.
Figure 3-4a



Square wave differential signal near the maximum transmission distance RS-422, 485, etc.
Figure 3-4b

The signal produced by a CANbus specific transceivers looks just like the RS-485. The most significant difference is what constitutes a differential change. The CANbus specification call for a 1 VDC change and RS-485 is 0.4 VDC. The differential changes are related to time, that is $\frac{d\Delta}{dt}$.

Both modulated and square wave signals can be impressed on top of slower modulated carriers and/or on AC or DC power sources. The slower signals (lower frequency) and the DC or AC power are the *carriers*. If the carrier is AC, the signal distance can be significantly increased. It's more expensive to impress a modulated signal on top of a modulated signal, but transmission distance can be miles without measurable distortion. This is the heart of cable TV.



Signal type versus distance
Figure 3-5

Signal type and transmission distance

Somehow I need to put the signal type and transmission distance into perspective. The problem is that a few successfully applied exceptions and many hypothetical exceptions allow for valid but endless arguments. Therefore the distance in figure 3-5 is general and represents a common usage today.

Broadband, baseband, carrier band

The transmission of the different signals (square wave or modulated) and the carrier, or lack of carrier, are commonly grouped into three categories:

Broadband:

A modulated or a square wave signal on a multiple modulated carrier. Distortions are minimal. Several carrier signals may operate simultaneously. Cost to transmit and receive is more expensive than for any other type. One-way direction of transmission per carrier frequency is permitted; therefore, if nodes are to send as well as receive on a single carrier, a ring concept is needed.

Baseband:

A single digital signal with no carrier. Distortion, grounding and distance are all potential problems. But this is the most economical method of transmission if the limitations are acceptable. Two-way transmission.

Carrierband:

This is just a one channel broadband. Digital or modulated signal on a single modulated carrier. It has all the advantages of broadband if only one channel is needed, but at a price somewhere in between that of broadband and baseband.

Broadband is too expensive for most device level applications. Carrierband is less expensive (components and wire) than broadband and might be justified on lengthy applications. Baseband is by far the most common signal for device level networks.

Data encoding

The word *encoding* is used at two levels, *bit* level and *byte* level. The most famous example of byte encoding is "... — ..." : this is Morse code for SOS.

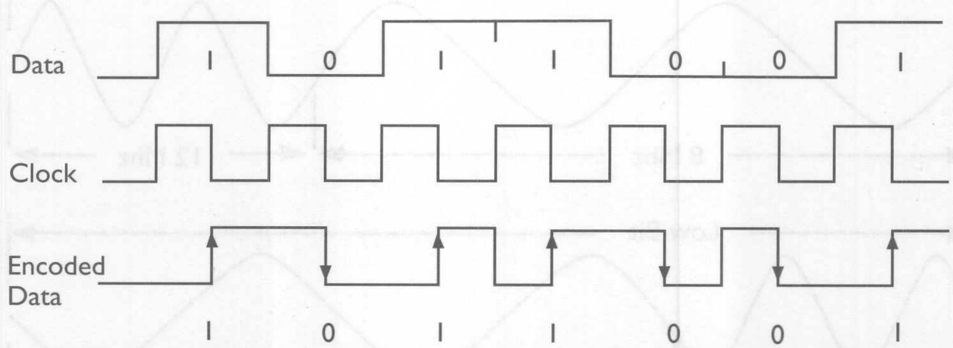
Today there are many common codes. Two of the most universal are: *ASCII* or *American Standard Code of Information Interchange* and *EBCDIC™* by IBM. In each case, a string of bits represents a character or symbol.

At the bit level there are several encoding methods. Of course, if a message is coded by the transmitter, it must be decoded by the receiver. In common data transmission jargon, the word encoding more-or-less covers both coding and decoding, unless we are referring to some specific receiver action.

Encoding puts a time reference to a signal to determine the boundary lines between bits. When we look at textbook examples of data, we see nice bold square waves with dashed lines in between bits that are always alternating high-low-high-low. But that is not what 10 straight highs look like 1500 feet down the busline. The signal is weak and distorted and does not have those neat little dashed lines.

The following drawing, figure 3-6, shows a common method used to put the time reference (neat little dashed lines) around a digital signal. It is called *Manchester* bit encoding method. For example, suppose we have a transmission data rate of 250 KBps. If we use a baseband digital transmission, that means a potential 250,000 bits per second are being communicated. If we use the common Manchester encoding method, we must have a clock (oscillator) that sends out a pulse at twice the data speed, or 500,000 pulses per second. A digital message using Manchester will have "Start of Message" bit(s). This synchronizes the clock. Because the clock is twice as fast as the data speed, every odd (1, 3, 5 . . .) pulse of the clock is the center of the data bit. In the past, we could lose time synchronization because of a long string of high or low bits. Also, minor deviations between the sender's clock and the receiver's clock could result in a loss of synchronization. Today, with high speed electronics, there are many ways to re-synchronize the receiver clock in the middle of a message.

Manchester works for a digital signal, but what about a modulated signal? The ideal modulated signal is a mathematically perfect sine wave. The sine wave gradually turns on, then gradually turns off, so there is no sharp change to start a clock. A sine wave operates on different frequencies. If we are communicating just binary information, then only two frequencies are required–one frequency for the high bit and one for the low bit (see figure 3 - 7). A bunch of highs in a row or a bunch



Manchester encoding is preformed by exclusive or-ing the data and the clock

Real-Time Control Networks Daniel T. Miklovic, 1993, Instruement Society of America

Manchester bit encoding
Fig. 3-6

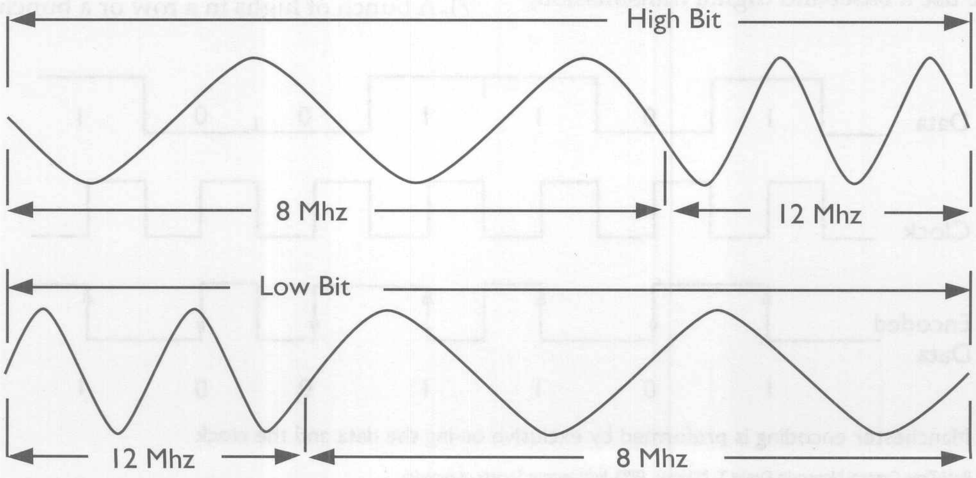
of lows in a row is just as much of a problem for the modulated signal as it is for a digital signal. Unless there is a time reference, 10 sets of high frequencies in a row could be mistaken by the receiver for 9, or maybe 11.

For a modulated signal, a common method called phase continuous frequency shift key modulation creates the time reference for the receiving node to demodulate (decode) the message. For example, the sender could use two frequencies: 12 MHz (12,000,000 full sine waves per second) and 8 MHz. The sender uses both for one bit–half of one and half of the other. But the order is important. For example, if the 12 MHz comes first, the bit is low. If 8 MHz comes first, the bit is high. The transition between the two frequencies in the first bit synchronizes the clock of the receiver. The transition of every bit essentially re-synchronizes the clock if there is a deviation.

Protocol

Every serial data transfer requires some organization. This organization is called *protocol*. For example, a dedicated signal between a mainframe computer and a dumb terminal has a simple protocol. The size of the message and the order of the message (bit importance/significance) is all that is needed. On the other hand, the protocol for TCP/IP for a WAN (Wide Area Networks such as Internet) is more complex. TCP/IP stands for Transmission Control Protocol/Internet Protocol. Protocol can be implemented with either *hardware, firmware, software,* or any combination. Bus developers must balance the complexity and cost of implementing the protocol against the versatility of the bus. Here is a case where one or two geniuses can get 20 pounds out of a 5 pound bag. Unfortunately, protocol is often turned over to a committee and the result is 20 pounds out of a 40 pound bag.

The upper end device level networks such as DeviceNet and Profibus have well developed protocols with provisions for more enhancements. Protocol should not be confused with network specifications. Network specifications define the color of a wire, the voltage, and the timing as well as the protocol. Specifically, the protocol establishes the format of the message and defines the actions expected when one node sends a message to another. An example of this is when a scanner module tells a node to report.



Phase Continuous Frequency Shift Key Modulation

Figure 3 - 7

Typical protocol information	Master-to-slave device level bus	High-end device level bus	Fieldbus for large distributed process control	WAN Wide area multiple networks such as Internet.
Network source address; necessary when more than one network are linked together.	N	N	F	Y
Network destination address; again for multiple networks.	N	N	F	Y
Time to live and time stamp; on large multiple networks, a message could live indefinitely in search of a non-existing network or address.	N	N	F	Y
Address of destination node.	S	S	Y	Y
Address of source node.	S	S	Y	Y
Operational flags.	S	Y	Y	Y
Message type flags.	S	Y	Y	Y
Size of the message or some part of the message.	N	Y	Y	Y
CRC or some form of checksum.	S	Y	Y	Y
Fragmentation flags; used when two or more transmissions are required to make a whole message.	N	Y	Y	Y

Key

N = This protocol feature is usually not included in this class of bus.

S = Some buses in this class have this protocol feature.

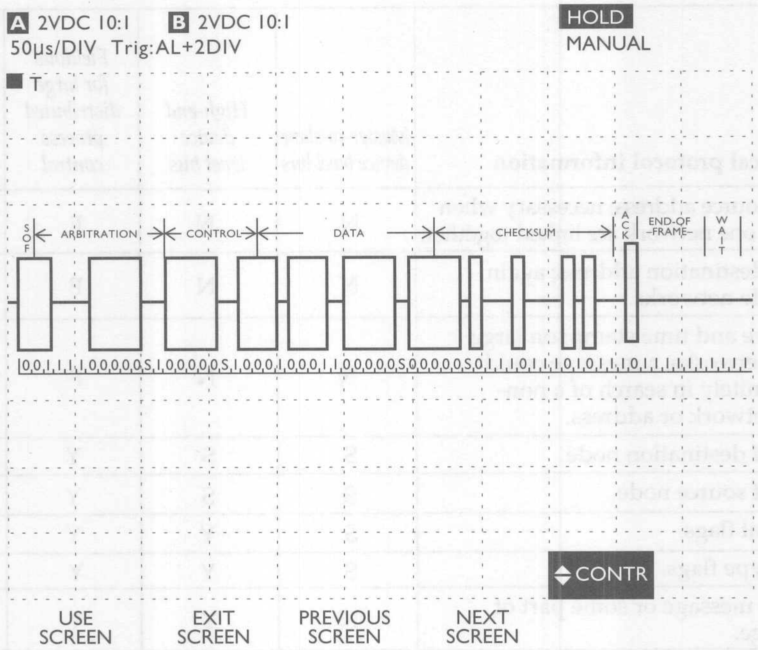
F = This feature will likely be incorporated in this class network as it evolves (it may have already be done).

Y = This protocol feature is usually included in this class.

Messages

What follows is an actual message coming from a device level node operating on a DeviceNet bus. This node is responding to a scanner-module that has polled it and told it to "report." The node is address 1. It is an 8 IN station and the 1st, 2nd and 6th inputs are ON. The station has no faults. The bits marked "S" are *stuff bits*. When there are five bits in a row of the same state, the transmitter inserts an opposite state bit, *stuff bit*, into the data. This bit is removed by the receiving node before the data is presented to the CPU for action. No *stuff* bits are used in the last 9 bits of the message and obviously no *stuff* bits are used when the bus is idle, because no one is transmitting. The *stuff bit* provides the clock with a hard change of state so it can re-synchronize during the message.

DeviceNet is based upon CANbus. So consequently part of the message has basic CANbus meaning and part has DeviceNet meaning. CANbus and DeviceNet operate at different levels. Appendix B goes into these levels in more detail. CANbus uses a concept of dominant and recessive bits. At first, it is hard to get used to it, because the dominant bit or "High" bit, is the "0" bit. And conversely the recessive or "Low" bit is the "1" bit.



Oscilloscope view of a message from an 8 point node using DeviceNet 1, 2, and 6 Bits ON, no faults

Figure 3-8

SOF	GRP	GRP	ID	Address	RTR	IDE	RO	DLC	Message State On/Off	Message Status (Diagnostics)	CRC Checksum and delimiter	ACK	End of Frame
0	0	1 1	1 1	000001	0	0	0	0010	00100011	00000000	0001110111010111	01	1111111

Protocol representation of a message from 8 point node using DeviceNet with 1, 2, and 6 Bits ON, no faults, and the stuff bits are not shown.

Figure 3 - 9

- The first bit is a start bit; it synchronizes the clock of the listeners. SOF means Start Of Frame.
- The next bit 0, identifies the message as group 1.
- The next 4 bits tell why this station is talking. Four ones means it is the slave's response to a poll command from a master.
- The next 6 bits are the address. The address is 1.

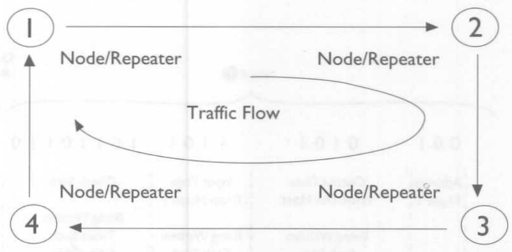
  Note: these 11 identifier bits are dependent upon one another for meaning. For an example: if the first two bits are 1 0 respectively, then we have a group 2 message and the address is indicated in the next 6 bits, not the last 6 bits of the arbitration field as in the group 1 message.
- The next bit is called RTR or Remote Transmission Request bit. Device Net does not use bit.
- The next bit is called IDE and is a CANbus indicators.
- The next bit RO is not used and is reserved for the future.
- The next 4 bits are binary to indicate the size of the data message in bytes. DLC means Data Length Code. Here 0010 means 2 bytes.

- Finally, here is the data message: The first 8 bits show that the sixth, second, and first bits are 1, indicating that the sixth, second, and first inputs are on. The next eight bits are 0, so none of them are faulted. In the data field, the most signficant data point is given first and the least is given last.
- The next 16 bits are the checksum and delimiter. This checksum is called CRC, or Cyclic Redundancy Check.
- The next 2 bits are the acknowledgment bits. The transmitting node sent both the bits as recessives, or "Lows". But the second bit is dominant, "High", so who sent it? In this case, it was the scanner module in the PLC. It understood the message and the CRC computed correctly, so it sent a dominant bit that overpowered the recessive bit. The transmitting node is listening as well as transmitting, and once it heard the second bit go "High", it knew its job was done. So it sent 7 recessive bits in a row to end the message, and then it stopped talking. If the second bit had not been "High", the transmitting node would have repeated the message. Another thing to note in the oscilloscope capture, is that the bit that was overwritten by the scanner module is at a slightly higher voltage potential than the other bits. The scope was attached to the data lines only a few centimeters from the node but several meters from the scanner module. The point is: this is a differential signal — it does not need a common reference voltage between the nodes.
- The next 7 bits are the end of frame.

A simple master-to-slave message is usually more concise than the CANbus message. It consists of the start and end bits, address bits, data bits, and checksum. Approximately 20% of the CANbus message is eliminated. But in fairness to CANbus, this comparison is unjust because with the appropriate protocol, CANbus can do more things than simple master-to-slave.

The master-to-slave and the CANbus message are somewhat intuitive. We could probably map a message evolutionary tree that goes back to the Telex™ and ultimately the orginal telegraph. That is why I put Messages early on in this tutorial. There are more complex ones such as those used on a ring type bus. We will look at the message here but you will probably have questions about the ring concept and its message management. Those questions will be covered in chapters 4 and 5 respectively, and we will look at 4 actual buses that use it in chapter 6.

The first important thing about about a true ring type bus is that every node on the bus is also a repeater (see figure 3-10). This is a not party line, where the nodes read and write a mutual data line. When a node talks on a true ring bus, only the next adjacent node hears the message. The message goes from node to node around in a ring. The data that is specifically for that node is read to set outputs. If the node has inputs, the node will write data for other nodes or a host controller.



On a ring type bus every node is also a repeater
Figure 3-10

Figure 3-11 shows a generic ring bus message and two nodes. What is implied is that there is a host that performs all logic and that there are several nodes on the bus. Each node has its own input and output data within the message. This generic example shows an input and output

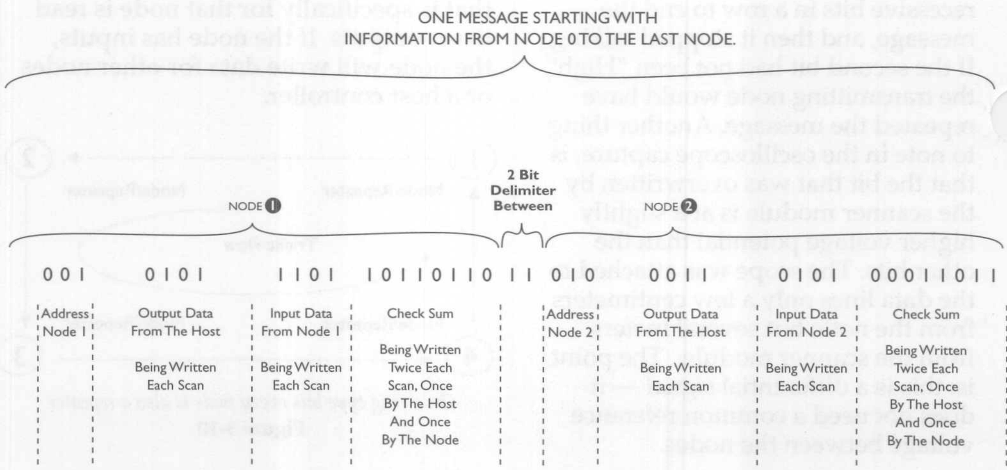
checksum for each node – a real world bus is more efficient and combines checksums. When there are multiple hosts the message can be considerably more complex.

Kick Some Bus Tires

Unfortunately, talking about bus talk is neither sexy nor glamorous. The signal type, the encoding method, and the framing of the message are important details. They are the meat and potatoes of data transmission. Think about the cable TV or the telephone line at home and the misuse it goes through. You may not make mistakes, but I do. I've shorted, I've opened, I've done about everything that's possible at home except for plugging it into the electric range receptacle. After proper reconnection it works again. A good bus system can come back from the dead. When you are making a decision about a bus, kick the tires a little. Open the data lines, short the data lines, reverse bias the data lines, but don't feed the 5 volt data lines with 24 VDC! Not yet!

Two electronic component companies have announced rugged CANbus transceivers that will be able to take any type of miswiring of the bus data lines or the bus power lines, as long as the bus power is a nominal 24 VDC or less. Check with your vendor for availability if this is important to you, and it should be, because a node with a blown transceiver is only a paperweight. At this time, I don't know of any EIA RS-485 type transceivers that can take a 24 VDC worst situation mistake.

Don't forget some noise. Try the bus around typical noise for your application such as drives, solid state motor starters, power converters, transformers, and walkie-talkies.



Ring type bus generic message

Figure 3-11

Chapter 4

The Bus Tour

This chapter is about the layout of the buses. Often there is a trade-off between the flexibility of a bus and the performance. Unfortunately, it tends to be an inverse relationship. The more free form the topology is, the less the performance. At the end of this chapter, I've included a short topic about a product type, called a gateway. It has considerable promise at the device level. This allows buses with different performance and usability to be coupled together.

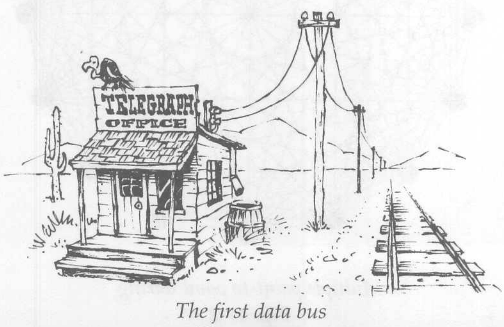
Bus routes and transfers

Topology is a term that applies to buses. It describes how the data lines connect the nodes together. If we have a WAN, it also describes the transferring devices that connect different buses and different segments together so information on one network can be shared with another.

The first bus goes back to the original telegraph, two data lines hanging on wooden poles going from telegraph station to telegraph station. (Maybe telegraphers didn't want to be called nodes in those days). This simple topology is called the *bus*. Most technical people feel uncomfortable describing anything with just one three-letter word. So we often hear *common bus, simple bus, or single trunkline bus*.

Due to the fairly long distances, power sources, and low technology of the wire, passive branching was limited. Often, *repeaters* were used because signals would become too weak after a certain distance. The first repeaters were the telegraphers themselves. They would receive and record a message on one segment, then repeat the message on another segment. Because the telegraph key and the sounder is analogous to today's pushbutton and relay, it is not hard to see how an electro-mechanical repeater would be implemented. The sounder (relay) would have a set of auxiliary dry mechanical contacts that could be wired into the next segment. The message could then be repeated in its original form across a continent. Both repeaters and amplifiers are used today on industrial buses. Repeaters recreate a fresh signal on the next segment. The disadvantage of repeaters (on some buses) is the inherent time delay (wait state) caused by the requirement of the original signal to rise above or drop below a specific threshold before the signal is repeated by a relay or a transistor. The advantage is that it is a good fresh signal.

The *amplifier* was not used at that time; linear transistors or vacuum tubes didn't exist. But today's amplifiers are common devices to boost a weak signal. The amplifier reads the old signal on one segment and creates a same shaped signal but amplified on another segment. The advantage of the amplifier is that it amplifies the signal in real time. (Technically there is a wait state with linear transistors; but it is small and even hard to measure.) The disadvantage is that it is not a fresh signal. It is the old signal with all the distortions, but amplified. The amplifier is most common on buses that use a modulated signal while the repeater is most common with square wave signals.



The first data bus
Figure 4 - 1

A *bridge* is another device that links two segments of the network. It is a little smarter than a repeater. Repeaters don't look at the message because they don't have the intelligence to understand anything about the message. They simply repeat the message. A bridge, on the other hand, does have a small bit of intelligence. You program a bridge to know the addresses on two adjacent bus segments. When a message originates in one segment and is destined for another segment, the bridge comes to life and repeats the message. When a message is destined within its own segment, the bridge is dormant. The addresses on the segments must be unique. The two segments must operate nearly the same. The nodes (telegraphers) must use the same protocol and data encoding method (Morse code).

The telegraph evolved in two directions. One was designed to go across a long distance, such as a body of water with undersea cable. These systems were expensive, requiring a tuned system with specific power sources. The other direction in which the telegraph evolved was land based. Routing features were more important on the land based systems. Railroads were the big users of the telegraph. If a telegrapher with the Union Pacific in Omaha wanted to send a message to a telegrapher at the freight house of the Denver Rio Grande in Denver, the message would most likely be repeated from segment to segment until it got to Cheyenne. In Cheyenne, the telegrapher would hear (Morse Code on the sounder) that the destination was Denver. He would then mechanically route the message to the Denver Rio Grande network, using his sounder as a repeater relay. So, we had *routers* in the early days of data transmission.

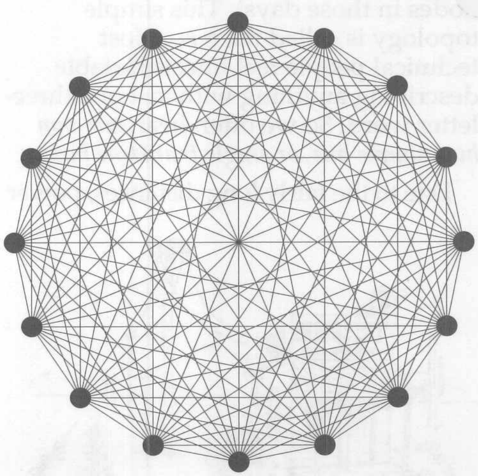
Routers do the same thing today. The destination network (the Denver Rio Grande Railroad) and the destination node address (telegrapher in the Denver freight house) are bits somewhere in the header of a message. The router reads this as the message comes in, and then electronically switches the message to the appropriate network segment. The router is probably the most important part of any network system with multiple networks. It eliminates traffic on portions of segments where the message has no destination. It also saves wire.

Multiple point-to-point

If the original telegraph system would have had to be designed as a point-to-point system, there probably wouldn't have been enough malleable conductive metal in the country to connect all points together. Two points are no problem, since only one pair of wires is required. The formula to connect a point-to-point system with all points connected is:

S = (p-1)+(p-2)+(p-3)+.....1
where: S = the total set of wires
p = points to be connected

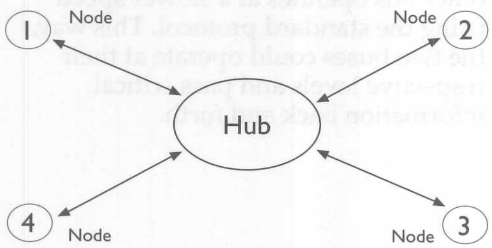
Obviously, this gets to be a big number fast as the following figure 4-2 shows. There are only 16 points, but there are 120 lines connecting the 16 points. If this were a two-way electric signal, there would be 240 wires.



Multiple point-to point wiring
Fig. 4 - 2

Star topology

Another wiring topology is called a *star* (see figure 4-3). This method was prevalent in the mainframe computer days. The big mainframe would be connected in this fashion to several peripheral devices such as tape storage, printers, and terminals. Today, the star is more of an idea than it is a reality. On paper, a hub (either passive or active) looks like a star, but internally, it is a common bus with many branches jammed together in a relatively small area.

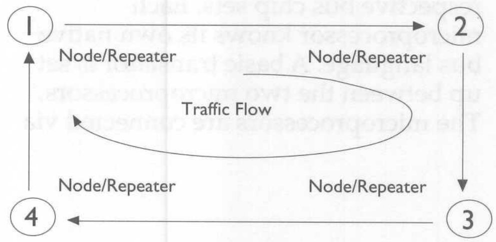


Star topology
Figure 4 - 3

Ring topology

The *ring* is another topology that is often conceptual. But, true rings do exist. On a true ring, every node is also a repeater. The information comes into a node, information that pertains to that node is read, new information is added and the message is sent on to the next node. With advanced protocol and additional hardware, the nodes can re-route the message if the ring is cut or a node-repeater dies. Look at Figure 4-4. Imagine a message flow in clockwise direction. Now imagine the ring being cut. When the message gets to the node-repeater just before the cut, it reverses direction and is repeated by each node back to the node just after the cut. The result is that the scan time has now doubled because the traveled length of the network has been doubled. It is usually an expensive system when all the hardware and wiring are tallied. It does lend itself to large (distance) networks and buses that use light rather than an electrical signal. More about this in chapter 6.

The definition of a ring is essentially a topology. But a method used to determine whose turn it is to talk is called *token passing*. In the mid 1980s, IBM came up with a technical, proficient network that was called Token Ring®*. Since then, many people believe that token passing and a ring topology are mutually inclusive, i.e., one does not happen without the other. Wrong! The next chapter discusses various ways for a node to get on the bus and talk.



Ring topology
Figure 4 - 4

Gateways

The last thing we'll talk about in this chapter is the *gateway*. We've looked at three different types of bus topology–the common bus (with or without branches), the star, and the ring. As you can see by now, buses differ considerably. The bridges and the router connect segments of the same type of bus. (Some routers can connect buses that have a similar protocol but different timing and electrical characteristics). The next step up is a gateway. A human equivalent may have existed at opposite ends of a Trans-Pacific telegraph cable. The receiver in Japan may receive a message in Morse code. The telegrapher would then translate it to Japanese, then encode it to a Japanese equivalent of Morse code, and then route the encoded translation on to the destination.

One person could have been a gateway, but it would have been rare. More than likely there were three people, a receiver, translator and a sender. Today's electronic gateways do

the same thing. They link two networks that differ in one or more of the following ways:

1. Electrical signal (voltage and reference)
2. Protocol
3. Signal type and carrier
4. Encoding methods
5. Network access (next chapter)

A common gateway design for connecting two dissimilar buses uses two microprocessors and two sets of the respective bus chip sets. Each microprocessor knows its own native bus language. A basic translator is set up between the two microprocessors. The microprocessors are connected via I/O lines and data passes back and forth between the buses.

Gateways will be popular at the device level. The most obvious use is to feed a device level bus into the next higher network level. ASI, one of the most limited (but economical) networks, has a gateway to link it into Profibus DP™. Another possible application would link similar buses where one bus with only a few nodes operates a high-speed application using a special extension of a base protocol and the other bus operates at a slower speed using the standard protocol. This way, the two buses could operate at their respective levels and pass critical information back and forth.

My turn to talk

The protocol is a small imbedded software or firmware code in each node on the bus. This code is conceptually divided into several areas. During the run condition there are two primary parts to it. One deals with the rules of getting onto the bus to talk; the other deals with the interaction between the nodes after the message has been transmitted. The latter can be (and often is) very application-oriented. Some parameters at this level may allow the user to customize part of the operations. It works similarly to the "Control Panel" settings in Windows™*. "Control Panel" allows the user to customize the mouse, printer, audio settings, and background screens, to name a few options. For a bus node, these customized settings could change the logic state of an input from N.O. normally open to N.C., or add an on/off delay timer to negate contact bounce.

The other portion of the protocol is very bus specific. It concerns the access method of talking on the bus. As you might guess, there are different types of access methods. Here are some analogies to help differentiate them:

Bus– CSMA/CD: unsolicited messaging

Analogy #1: It's a college campus and two undergraduate students of similar dress, verbal abilities, and temperaments meet in a situation that may require a conversation. Possible results:

- Undergraduate A speaks first, and B waits until A is finished.
- Undergraduate B speaks first, and A waits until B is finished.
- Both start to speak at the same time (see illustration); both stop after finishing the first sentence. Both realize an information collision has occurred because they were talking simultaneously. The students resolve the problem by applying the rules devised by their college to correct such collisions. According to college rules, students are to take one strand of their hair at random, measure it, multiply it times their GPA, and then convert the answers to nanoseconds. This number tells them how long they must wait between utterances.

In the bus world this is called CSMA/CD operating in an unsolicited messaging mode. CSMA stands for Carrier Sense Multiple Access. This means the node will not talk if some other node is talking, but has equal right to talk if the busline is clear. The /CD Collision Detection means that both listen for a data (electrical signal) collision while they are talking. If there is a collision, the two or more nodes use an algorithm that will create a random time delay before a node will speak again. The time delays are almost never the same for the two nodes. Even if it does turn out to be the same once, it is nearly impossible for it to occur twice. The messaging type is unsolicited. This is also called event driven, meaning that if there is a change of state at the node, it will want to tell all about it.

Bus–CSMA/BA: unsolicited messaging

Analogy #2: Now suppose one of the students graduates and joins a structured old law firm. Both the founder and the recent graduate enter an elevator together. A conversation may take place. Possible results:

- The founder starts speaking and the recent graduate waits until the founder is finished before speaking.
- The recent graduate starts speaking and the founder listens until the recent graduate is finished before speaking.
- Both start speaking at the same time; but, because the founder is employee #1 at the firm and the recent graduate is employee #135, the founder has priority. The rule at this firm is that the person with the higher employee number must stop speaking and let the person with the lower number finish.

The equivalent bus is called CSMA/BA operating in an unsolicited messaging mode. The CSMA is the same as before but the */BA* is different. This is *Bitwise Arbitration*, to determine who has the right to continue talking. The node with the lowest address wins and continues talking. The higher numbered address has to wait until the bus is clear to talk. Again, the messaging is unsolicited.

Bus–CSMA/xx: solicited messaging

Analogy #3: The other student graduates, leaves college and goes into the military and is standing in ranks in basic training. A drill sergeant walks up to the trainee. A conversation may take place. The only possible result:

- The drill sergeant yells at the trainee to report. The trainee reports and then stops talking.

This is analogous to any CSMA bus, but the bus is operating in a solicited messaging mode. Although a node on a CSMA may have the ability to initiate a message, the rules are very explicit: "Don't talk unless told to."

Bus–Master-to-slave: solicited messaging

In the master-to-slave bus, both the master and the slaves are nodes. When the master calls out the slave's address, the slave does not even look at the bus, it just reports. This works because the master takes full control of time and the operational sequence of the bus. There is no bus traffic except when the master initiates it. On a master-to-slave bus only the master can talk to a node. Nodes do not listen or react to what another node says. The slave nodes are obviously operating in solicited messaging mode.

Bus – Token passing: solicited messaging

The last method of talking on the bus called *token passing*. It works like this: One network master starts the polling (report) sequence. Once the sequence has started, each node then follows the predetermined sequence and reports when it is time to report.

The following scene may help illustrate the concept:

> A teacher is passing an eraser around a room of grade school students. According to the rules, only the student with the eraser may speak. After that student's predetermined time to speak is over, the student passes the eraser on. If the eraser is dropped on the floor or if a student isn't paying attention when the eraser is passed, it is the teacher's job to step in and restart the eraser-passing. The job of the master in the token passing bus is like that of the teacher; it controls the bus — especially the timing.

Some people break down the token passing bus into three groups:

- Bit shift
- Byte shift
- Message shift

The main difference is the size of the message. The bit shift has the smallest message and the message shift has the largest. Also there is a correlation of size to the complexity. In the next chapter we will look at examples of real world token passing buses.

Advantages or disadvantages of solicited or unsolicited messaging

A CSMS/BA with solicited messaging looks a lot like a master-to-slave bus with the same type of messaging, but when the CSMA/BA bus is operated with unsolicited messaging it is completely different. In this case, messaging mode is more important than the bus type.

Advantages and disadvantages are not universal from a small bus in a copy machine to a huge network such as Internet. Since this tutorial is about the device level we will weigh the pluses and minuses with respect to a device level network.

Advantages of solicited bus traffic control

It is deterministic. Some fairly accurate methods can be used to determine the scan rate and this rate is repeatable. It may be as simple as counting the number of nodes, multiplying this number by a constant and then adding an overhead constant. This makes it possible to calculate how long an input must stay "On" or "Off" before that condition is communicated to other nodes or a master.

Shut-down of the output station is programmable. An important feature that may be added to output nodes on a solicited network is the ability to recognize loss of communications. On a solicited bus, there is constant traffic and on most solicited buses the output nodes are polling each scan for health status. Therefore, the output nodes can determine if they are still able to listen and understand. A program may be added to the output node to cause it to shut down in an orderly fashion. For example, it might be desirable to turn "OFF" the outputs if the output station has not heard anything on the bus for more than 20 milliseconds.

Nodes are solicited sequentially. Some order is followed in calling the nodes to report. The order of event happening and the reporting of the information is not an absolute time-line. For an example, a change, say at node 4 may happen before a change at node 2, but both may be reported as equals during the same scan. This is not ideal, but this is the way it happens with input cards in a PLC rack today. It is what we are used to.

An attendance check is possible for each scan. This feature is implemented in one way or another on most solicited buses.

Disadvantages of solicited bus traffic control

It is difficult, but not impossible, to prioritize nodes. If one node's information is needed more often than a complete scan rate allows, the bus must be broken down into smaller segments, or software must be used that allows selective scanning. By the time you read this, selective scanning will only require a few key strokes with some of the more advanced buses and software.

Unchanging states are repetitious for each scan.

Advantages of unsolicited (event driven) bus traffic control

Only change-of-state messages are on the bus. Bus traffic is less than it is for solicited operations.

CSMA/BA buses run in an unsolicited mode, and are well-adapted to prioritization.

Information is sequential if the bus is lightly loaded.

Disadvantages of unsolicited (event driven) bus traffic control

Any node, gateway or interface to a PC or PLC that does something with the information on the bus (such as set an output or transfer data) must do so at the maximum data rate without an excessive wait state where a message could be lost.

Typically input nodes must be detuned in some manner. The electronics of discrete input nodes are fast enough to see the contact bounce on mechanical contacts, so on-delay and off-delay timers should be incorporated into the electronics. If the timers weren't put in, the node could potentially overload the network by reporting contact bounce. Analog inputs may be even more of a problem if the resolution (accuracy) is high. Instead of reporting every fluctuation, the resolution must be detuned so the analog input does not dominate the network.

It is not deterministic. In normal run mode, this disadvantage may not be a big problem on certain applications. But most of the industry is used to the idea of a scan rate from using PLCs. Machine builders have a good idea how long it takes a sensor to see a target, process the logic and turn an output "ON" or "OFF." Some buses that use solicited messaging can predict reporting. The prediction is a statistical model and not real world absolute.

Nodes are not able to distinguish between low bus traffic and no bus traffic. Are there no messages because nothing is happening or because there are failures? This inability to distinguish between the two conditions limits features that might otherwise be added to the nodes. Such features include diagnostics and LEDs to indicate communication problems or programmed shut-down of output nodes if communication is lost.

The outputs will obey or else

Another way to look at this topic is what do output nodes do. Do they tak output instructions from only one other node? Or can they take instruction from more than one node?

If the output node takes commands from only one other node during a run condition, and that commanding node has the ability to command other nodes, then we have *master-to-slave* interaction.

Another interaction is <u>*limited*</u> *peer-to-peer*. An example of this may be a network with 16 one channel input nodes and 16 one channel output nodes. The inputs and outputs are paired-up. An output node receives data from only one input node and that input node does not provide data to any other node.

Only a few buses at the device level

support *full peer-to-peer,* but this could change as enhanced protocols are implemented. Full peer-to-peer means that an output node may require the information from several different sensing nodes for a logic solution. (The output node is beginning to sound more and more like a very small micro PLC.)

Explicit messaging

One last thing is something called explicit messaging (then we can stop defining the bus). Actually a master writing to a slave is an example of explicit messaging. Somewhere in the message from the master is the destination address, then the data. This is the only way a master-to-slave bus can work.

Explicit messaging is often used when initializing the nodes. Some parameter is typically down loaded to a node. It may be an address change, or it may be a request for a node's serial number or who manufactured the node.

Explicit messaging may also be use by an input node that thinks it is a micro PLC, and is commanding output nodes. This hasn't happened often yet, but it will happen not very far in the future.

Summary of Bus types

This is the only chapter that will have a summary. If you are like most bus novices there still is something that doesn't quite make sense. The engineering mind is at a disadvantage when thinking about the bus types.

When we hear about variables and types we think we have to come up with a complete row and column matrix. The problem with buses is that the matrix has holes and some terms have more than one meaning. Nevertheless, I will give it a try on the page. I will make the media access method the fixed element. The parameters will then be the variables.

Comments

These new buses commonly have input and output nodes that use microprocessors that are as powerful as the CPU used in the first generation of PCs back in the early 1980s. And for a few bucks more, the nodes can be more powerful. As you know, with a PC you can run spreadsheets, databases, word processors, and games – it is a matter of software. The bus types, the messaging, and the interaction may some day become just a selection made with a mouse. The digital buses can almost do that today. What is holding them back? Simple things like color codes of the wires and pin-outs of connectors. When you get down to it, almost all digital buses use just two different types of transceivers. One is a variant of the other. These transceivers don't care what color the wire is. They don't care about the one-up-manship on the wire gauge of the data lines. What is important is the dielectric strength of the insulation, the capacitance between the data lines and the speed of signal propagation.

Media Access method	Collision avoidance/ detection	Messaging (When do nodes talk)	Comments	Buses in Chapter 6 using this technology
Master controlled	Not applicable	Explicit: master writes to output node. Explicit: master sends a report request (poll) to input nodes. Solicited: slaves respond to request from the master.	Traditional master-to-slave bus.	Sensoplex® ASI Profibus DP Bitbus
Token passing	Not applicable	*Implemented with a single network-master that manages the token passing and a host that determines all logic.* Explicit: host writes output data when the token permits. Solicited: input nodes send data when the token permits.	Functionally the same as master-to-slave, all time and sequence is controlled by the network-master and the host performs all logic.	Interbus-S Sercos Seriplex mode 2 Beckhoff Light Bus
		Implemented with a single network-master that manages the token passing. Input nodes have exclusive corresponding output node. Solicited: input nodes send data when the token permits.	This is referred to as a limited peer-to-peer token passing bus.	Seriplex mode 1
		Implemented with a single network-master that manages the token passing. Output nodes listen for information from one or more input nodes. Output nodes are logic capable. Solicited: input nodes send data when the token permits.	This is referred to as a full peer-to-peer token passing bus with broadcast input data.	GE Genius I/O
		Implemented with a single network-master that manages the token passing. All messages include source and destination addresses. Solicited/explicit: nodes send data to other nodes when the token permits.	This is a full peer-to-peer token passing where any node may have logic capabilities.	ArcNet
CSMA	Bitwise arbitration Collision detection	*Implemented with a single master and all field nodes are slaves.* Explicit: master writes to output node. Explicit: master sends a report request (poll) Solicited: slaves responds to request from the master.	In the run mode this is functionally a master-to-slave, although the CSMA allows options during start-up	DeviceNet first implementation DeviceNet also has a *strobe* message that tells multiple nodes to report
		Implemented with a single network-master that manages start-up chores, but then the nodes access the bus when there is a change. The host performs the logic. Explicit: master writes to output nodes. Explicit: master checks for attendance. Solicited: slaves report attendance. Unsolicited: slaves report a change.	This implementation is similar to on-board automotive applications. It is best called an interrupt driven CSMA bus with a host. This bus is not deterministic nor is there a scan. So there is not an attendance check every scan.	SDS, DeviceNet rev. 1.4 will include change-of-state
		Implemented with a single network-master that manages start-up chores, but then the nodes access the bus and freely exchange data. Logic may be performed by input, output or multiple masters. Explicit: an input node may write to an output node. Unsolicited: input node may report a change. Explicit: a node may request information	This may be the ultimate in versatility, but it is not deterministic. Again there is no scan so there is no attendance check per scan.	Lonworks

Chapter 6

A Choice of Device Level Buses

Getting on Real Buses

The previous chapter talked about characteristics and features of bus types. The language was general, but usually optimistic in reference to the potential of a bus type. In the real world, however, some of these potentials have not been realized. This chapter talks about popular real-world buses. This list includes buses that already have made an impact or at least have vendors with sufficient marketing and production power to make an impact.

You'll notice as I describe these buses that I qualify their features. Often, we are dealing with what might be called a *gross* feature, for lack of a better term. However, the *net* feature is what really matters to the user. For example, we hear about bus speeds from 9.6 KBps (9,600 bits per second) to 12 MBps (12,000,000 bits per second). This says nothing about the net information speed. The "ON - OFF" information can be less than 1% of the bus speed. I have seen the net information speed at less than 0.05% of a 250 KBps bus speed. Obviously, in this case, doubling the bus speed leads to a slightly better net information speed of a whole 0.10%. Wow! On the other side, I've seen a master-to-slave using a modulated signal approach 25% of its bus speed, and a token passing ring bus could theoretically hit 58% when fully loaded.

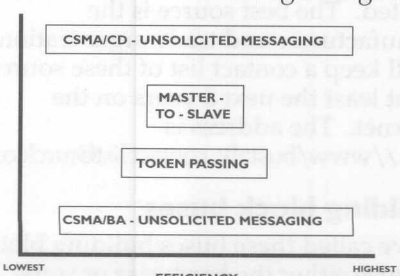
Obviously there must be a way to qualify net information speed. This could be a startup reason for a new magazine, similar to what PC Magazine did for the personal computer.

Figure 6-1 shows a general relationship of bus type to net speed. A CSMA/CD bus operating in an unsoliticed messaging mode can be the least efficient and the most efficient – that is, either the slowest or the fastest speed depending upon the application. During startup of a large machine, the network may become overloaded as the nodes compete for the bus to report changes. The competition for the bus will result in repeats due to collisions. Therefore, during momentous change-of-state, it can be inefficient. During low active periods (the right side of the bar) it is very efficient, because only the changes are reported and collisions are negligible.

The master-to-slave or any CSMA/xx bus operating in a solicited mode has a fairly narrow range. Half of the bus traffic is polling by the master and half or more of that message is overhead (start of message, address, checksum, etc.). The response is the other half of the bus traffic and again usually more than half is overhead.

Token passing buses are usually more efficient than master-to-slaves. Inefficiencies do happen when the actual bits/bytes/words don't match the intended message size. Also destructive electrical noise can be a significant efficiency problem for a true ring type bus using a traveling packet of information. The true ring type bus is where every node is a repeater. One altered bit will invalidate the checksum, resulting in an entire traveling packet of information from as many as 256 nodes having to be scrapped. An altered bit on other bus types will cause the loss of data from just one node.

The last bar in the figure is a CSMA/BA bus operating in an unsolicited mode. The right side of the bar is just like the CSMA/CD bus. The inefficient shaded area at the bottom represents nodes that are not setup to eliminate contact bounce or minuscule analog changes.



Relationship of bus type to net speed
Figure 6-1

Another significant factor that affects the efficiency of a bus is the utilization of the potential data message. Sending 60 bits and getting back 60 bits to indicate one "On" or "Off" condition is not very efficient. But increasing the I/O count per node to 4, 8, 16, or 32 improves the efficiency beyond what is usually available by pushing the hardware to higher speeds.

The last speed issue has to do with the hardware. I've put some of the buses on a scope and seen massive idle time when there should be traffic because the host interface node is either talking to the host CPU or cranking through its program.

My ranking of the speed issue relates to the following factors from the most significant to the least:

1) Host interface node
2) Utilization of data message
3 tied) Bus type
3 tied) Bus speed

A look at the buses

We will look at five building block buses, six general purpose open buses, two specialty buses and two light buses. The first five chapters have been leading to this. None of the buses can be plugged into all different PLCs and industrial computer platforms. The compatibility with the controller you use is an important factor in selecting a bus. The available interfaces and gateways are changing and expanding quickly. I could list the gateways and interfaces in this tutorial but they would be obsolete before this tutorial is printed. The best source is the manufacturers and trade organizations. I will keep a contact list of these sources for at least the next 3 years on the Internet. The address is http://www/bustalk. contact.list@turck.com

Building block buses

I've called these buses building blocks because either the hardware or some level of protocol has been left for the vendors, systems integrators, OEMs and/or user to finish. The first EIA RS-485 is actually an electrical specification with wide open protocol. The last, ArcNet, is a building block for some buses. It has add-on protocols that finish the bus to the point that only a little application customization is left to do.

EIA RS-485

EIA RS-485 is a standard that describes the electrical characteristics of receivers and transmitters connected to a transmission media. The transmitters are the signal generators and, of course, the receivers listen to the communications. The standard specifies a differential signal betwen -7 to +12 volts.

The standard also specifies short circuit current to a source power supply and a rise rate with respect to time of the signal from one logic state to another. The specifications state the maximum number of loads is 32. That means we could have 1 transmitter and 31 receivers, but most often 32 combination transmitters and receivers called *transceivers* are used. To the line, a transceiver appears as one load, not as two. EIA RS-485 does not specify the wire or protocol. In bus jargon, EIA RS-485 is the specification of the hardware that *accesses the transmission media*.

Another thing that is noteworthy about EIA RS-485 is that interconnection between transmitters, receivers and wire is balanced with respect to ground. The transceivers, therefore, detect a difference between the two wires, not an absolute difference between the wires and ground. This has two benefits. First, on longer runs, the ground from one point to another is seldom the same; this method doesn't care. Second, a ground system is usually more susceptible to noise, especially in a factory environment. None of this is very exciting today, but it was when communication evolved from an unbalanced system using RS-232 and RS-423 to the balanced RS-422 and RS-485.

EIA RS-485 can be a quick and inexpensive bus for lab work where the users can write a small dedicated communications program. But it falls short as a general purpose device level bus because of the lack of universal protocol.

Bitbus®

This building block bus is from Intel. This will be a quick honorary paragraph because Intel held an end-of-product sale for this product during the fall of 1995. The chip was based upon the old NMOS technology which consumed considerably more power than today's CMOS. It operated at three speeds: 62.5 KBps, 375 KBps, and 1.0 Mbps using a maximum of 10, 3 , and 0 repeaters respectively. The number of nodes depends upon the speed, again with 256, 96, and 32 respectively. It uses standard EIA RS-485 transceivers and signal levels. It has CRC error checking and correction protocol. It does not have duplicate address checking. It is master-to-slave with a single master. It is also an IEEE 1118 standard. A 9 pin sub D shell connector was defined but no industrially hardened round connector was specified. The data message is 13 bytes. There are quite a lot of these buses in the field, but unless substitute silicon is found Bitbus® is obsolete.

CANbus

This is another building block bus. It uses a square wave digital signal. In its plain format, the media access method is CSMA/BA (see **Bus Traffic** in section 5). It has built-in error detection and correction. Attendance checking is not built into CANbus but all the data is there for additional protocol resolution. The built-in CANbus protocol allows for a variable data message from 0 to 8 bytes (8 bits per byte). Additional message length can be added with advanced protocol.

Today's products incorporate CANbus within microprocessors. The microprocessors also operate the node's input-output functions and additional features such as local indication LEDs. The microprocessor can be programmed with advanced protocol (see **DeviceNet and Smart Distributed System** later in this chaper). CANbus has no inherent speed restriction. Speed is a function of the microprocessor, the transceiver, the transmission media and the transmission distance.

CANbus has proven itself rugged and forgiving in automotive applications. The semiconductor industry will sell about 40 million microprocessors with CANbus interfaces in 1996 making CANbus the most common non-office bus.

LonWorks

Echelon's LonWorks, based upon the Neuron™* integrated chips is included in this section of building blocks because it does so many things. The Neuron chip has 3 microprocessors: the MAC processor, the network processor, and the application processor. Initially the MAC processor accessed in a CSMA/CD manner. An advantage of having a dedicated processor for bus access is that any access method is possible as long as there is sufficient demand to warrant development of a new code. A master-to-slave access has been added. The operation of this processor is proprietary.

The network processor handles such things as addressing, authentication, background diagnostics, network management and routing functions. This processor is also proprietary.

The application processor is open to vendors, system integrators, OEMs and users. LonTalk™* is a software package used to program the application in this processor.

Another reason for putting LonWorks in this section as a building block bus is that Echelon has not specified a transmission media. This job has been left to user and vendor groups. Echelon does market transceiver packages that couple the Neuron chip to common two wire media.

The overhead of three processors hindered LonWorks initially on high speed machine tools. Higher speed Neuron ICs have increased the throughput. The horsepower of the three processors has made the Neuron IC favorable for building environmental control and security systems. Process control may also be possible using the new master-to-slave access with deterministic scans.

ArcNet

ArcNet is an enduring token passing bus introduced in 1977. It was designed from the beginning to handle communications, so the host would be free to run the users' applications. ArcNet was heavily used in office networks until the mid 1980s. Since then the CSMA/CD buses have dominated. ArcNet allows every node time to talk to every scan. This is usually wasteful in an office environment, where a PC may be in a local operation for minutes or hours before a text file is sent to a network printer. This disadvantage in the office is a tremendous asset in process, batch, or machine tool control.

ArcNet is may be the most open bus product with the exception of EIA RS-484. The intellectual property of ArcNet is public domain. The only thing proprietary about ArcNet is the trademark by Datapoint Corporation.

ArcNet has been around for so long that there are adapters for any type of transmission type (broadband, baseband, carrier band) or transmission media (fiber optic, coax, twisted pair, etc.). It has become the building block of several industrial networks in distributive control systems and machine control. By copywriting a small amount of additional protocol code, this open bus can be essentially closed. Then, the closed bus is given a new name so the users won't think they're getting an office bus.

ArcNet is not the right network for a single bit input or output node, but it does such an excellent job at transmitting medium (128 bits) to large (4096 bit) messages, that I question the merit of reinventing this level of bus. New hard and soft products from SMC®* of Hauppauge, NY make ArcNet easy to use and flexible.

There are many low cost interfaces to PCs. In the standard AT-ISA structure the cards are selling for less than $100.00. ArcNet has the horsepower to be an ideal collector bus being fed by lower device level buses via a gateway.

General purpose open buses

Most of the information following has been derived from the manufacturer's or association's own specifications or literature. So that there is little argument, I've shown the node count as the maximum (physical, virtual, and/or implied) number of nodes on the bus. Yes, the available addressable nodes for the user are less, depending upon the system and how it is implemented. Usually there is one taken up by some type of master, address 00 or 01.

Another is often lost to the last possible field address. The manufacturers of these nodes use this last possible address as a default. This address could be used on a bus as an active address, but it is convenient to keep this address open to bring in, check-out, and re-address new nodes.

Another address is typically reserved on some buses for an active monitoring tool. This may be a laptop PC or hand held monitor. If it can emulate a field node, force an I/O point, or change the address or parameters of a field node, then this monitoring tool usually needs an address.

ASI

ASI is an acronym for Actuator Sensor Interface. This may be the most high-h bus of the group, high-tech like ssue paper. A lot of technology has gone into the development and design, but for the user, the application is simple because almost everything about this bus is fixed. About the only thing the customer gets to choose is the color.

The most unique feature about this bus is that it's designed to operate with unshielded UTP (UnTwisted Pair) wire. Both power and data are carried on this single set of two wires. The topology is the standard bus with unlimited branching and drops. The only requirement is that no more than 100 meters are used in a bus segment. Impossible around industrial noise? No. This bus uses a high-tech signal, a master station that controls time and bus administration, a fixed bus speed, a fixed message size, and a limited bus length.

This high-tech signal is called APM, Alternate Pulse Modulation, with Manchester II coding and decoding. The modulated signal is a sine^2 wave. This type of signal radiates very little noise that could affect other equipment. With filtering, it is fairly immune to external noise. Both master and slave are capable of detecting errors and asking for repeats. The distance and speeds are fixed. And because it is a master-to-slave bus, there is no need for customer variable protocol. In fact, most of the protocol is built into the ASI hardware.

Bus type	Master-to-slave, explicit and solicited messaging, single master
Total number of nodes	32
Bus topology	Straight with unrestricted branches
Physical distance	100 meters all trunk branches and drops
Transmission media	2 wire untwisted, unshielded, flat IDC, (Insulation Displacement Cable)
Transmission signal	APM with Manchester II encoding
Input bits per node	4
Output bits per node	4
Speed	167 KBps, 5 msec to read/write all nodes; this is approximately 1 msec per scan used by the interface/gateway and 0.125 msec per node
power	2 Amps using same 2 wires as the data signal
uplicate address detection	Yes, two nodes with the same address will result in "out of profile" signal
Attendance check per scan	Yes, an attendance list is programmed in master and checked each scan.
Error detection	Yes, single parity bit.
Error correction	Yes, master or slave can request up to three repeats before faulting
Address setting	Off-line with hand held programmer or dedicated master. On line via a master and a reserved newcomer default address that is changed to a run address by the master. No dip/rotary switches possible.
Node parameter programming	The 4 output bits can be used to change node parameters, but these bits then are not available as actual outputs. Programming is then essentially on-the-fly via the host.
History	ASI was founded in 1992 by a consortium of mostly German manufacturers. The original group consisted of Siemens, TURCK, Festo, P&F, IFM, and others. Today, the ASI membership is open to any control equipment manufacturer or supplier.

Summary: This bus will develop as a small PLC bus and a sub bus to higher level buses. At the time of this writing, the small PLCs are mostly Siemens. Gateways have been developed for DeviceNet, Profibus, and Interbus-S. Gateways are technically possible for Genius I/O, SDS, the fiber optic buses as well as buses based upon Lonworks or ArcNet (some may already be finished). In the future, a two wire connectorized bus version will be marketed with standard quick disconnects in addition to the IDC connections.

DeviceNet

DeviceNet is based upon CANbus using the 11 bit identification standard. This 11 bit identification is broken down into 5 bits for the type (32 flavors) of messages and 6 bits for the MAC ID (64 addresses). In its native form, CANbus is a CSMA type of bus, but additional protocol has been written to make this a master-to-slave bus. Revision 1.4 of the DeviceNet specification will further define a change-of-state operation that will be more like the native CSMA/BA operation.

The variety of CANbus processors and software make this a viable bus. Over 40 million CANbus chips will be made in 1996, making CANbus base networks the most widely used non-office network. There are several hardware and software tools that have been developed for CANbus in the automotive industry that can be ported to DeviceNet applications.

This digital signal bus has proven reliable even in the engine compartme... on automobiles and trucks. The only negatives may be the distance and the limitations on branching. Because of the bitwise arbitration used during power-up and certain messaging types, this bus does not lend itself to repeaters. If in a large system, where both node count and distance are needed, DeviceNet would make an ideal bus to possibly feed into another higher level bus. Significant branching could also be achieved with active hubs. This method would cost more but may be worth it when branching into areas where the physical mishaps could damage the bus power and data lines.

DeviceNet

Bus type	CSMA/BA, master-to-slave, explicit and solicited messaging. Unsolicited messaging, multiple masters, and peer-to-peer was introduced in early 1996 by the Systems SIG group of the ODVA™ for approval, conformance review and implementation in 1996 and 1997.
Total number of nodes	64
Bus topology	Straight with restricted drops
Distance	500 meters full trunkline, 6 meter branches @ 125 KBps 250 meters full trunkline, 6 meter branches @ 250 KBps 100 meters full trunkline, 6 meter branches @ 500 KBps 156 meters accumlative distance of the branches @ 125 KBps 78 meters accumlative distance of the branches @ 250 KBps 39 meters accumlative distance of the branches @ 500 KBps
Transmission media	Full trunkline – 2 wire twisted shielded cable with 2 wire bus power cable and drain wire. Thin trunkline – same as full above but with lesser wire size which is more economical and easier to install.
Transmission signal	Square wave digital with NRZ (Non Return to Zero) encoding.
Input bits per node	64 bits (allocated in bytes) standard in polling mode, larger fragmented messages are supported.
Output bits per node	64 bits (allocated in bytes) standard in polling mode 24 bits in explicit messaging. Fragmented messages are also supported.
Speed	125 KBps 250 KBps 500 KBps
Bus power	8 Amp full trunkline 4 Amp thin line
Duplicate address detection	Yes, nodes announce address on start-up and all listen. If a duplicate address is heard, the duplicate nodes will not advance to run mode.
Attendance check per scan	Yes, a list is programmed in the interface and checked.
Error detection	Yes, CRC
Error correction	Yes, nodes that detect errors signal the sender to repeat.
Address setting	Off line via hand held programmer or with a dedicated interface and host. On line via the interface master using a reserved newcomer default address that is changed to an application address. Dip/rotary switches are optional.
Node parameter programming	Can be very extensive to include drive/rotational and instrument parameters
History	The early days of DeviceNet were essentially Allen-Bradley. In 1992, Allen-Bradley started to share information and invite not only strategic partners, but direct competitors to become DeviceNet members. DeviceNet was released at the ICEE show in Chicago, March 1994. Then a year later, Allen-Bradley turned DeviceNet over to ODVA, Open DeviceNet Vendors Association.

Summary: This bus has all the ingredients for success: available silicon, an early sponsor (Allen-Bradley) with deep pockets to get it through development stage, network management software, and now a strong vendors group (ODVA) to refine, promote and evolve the product. It has strong backing from control manufacturers in the USA and Japan, the number 1 and 2 world economies.

Interbus-S

Interbus-S is perhaps the most unique bus in this group. It is a true token passing ring. Although it may cost a little more to implement, Interbus-S is a good candidate when a large (distance) system is needed with a fast and deterministic speed. Interbus-S is actually two buses. The primary bus is known as remote bus, with a maximum of 64 nodes. To a programmer, Interbus-S looks like a word (16 bits) shift register, covering a distance of up to 8 miles. During a cycle, each node receives a message from the previous physical node, takes what it needs, adds local information, and then transmits it on to the next node. This method adds some complexity to the protocol but each of the nodes can be up to 400 meters apart. The signal conforms to standard EIA RS-485 and is transmitted at a modest 500 K Bits/sec but the effective rate due to the low overhead makes this message shifting ring one of the fastest device level buses available. It is deterministic like any token passing bus.

The second bus is known as a local bus. This secondary bus permits a 10 meter maximum drop from the 64 remote bus nodes. This bus is not isolated and uses the I/O lines directly from the Interbus-S ASIC. Within the ASIC, the inputs are conditioned and the outputs use internal 12 mA drivers.

As with any token passing bus, a network master is required to administer the network. The network master is also the interface to a host or gateway into a higher level network. The Interbus-S protocol has a parameter initialization mode. Like all higher device level buses, the network can transmit parameters to initialize drives, instruments, and other higher level I/O.

Interbus–S

Bus type .... Token passing, explicit and solicited messaging, output field nodes are written to only by the master. See Chapter 3, Messages for a brief review of type data message.

Total number of nodes .... 64 remote bus and 192 local bus nodes

Bus topology .... Ring, but a physical straight bus. The message starts at the master and goes from remote node to remote node until it gets to the last node, then returns to the master via each node but, on the return trip the nodes are simple repeaters. The cable from remote node to remote node has a complete set of outbound wires and inbound wires. So the cable is a straight bus, but the two sets of wires in the cable make it a ring.

Distance .... 400 meters between remote bus nodes and 10 meter local bus drops, total distance of the remote bus is 25,600 meters

Transmission media .... Standard remote bus 3 pair twisted wire with shield and drain. Local bus requires 5 twisted pair wires and shield drain.

Transmission signal .... The remote bus uses a square wave digital signals, EIA RS-485 standard.

Input bits per node 16

Output bits per node 16

Speed .... 500 KBps

Bus power .... Optional for sensor/actuator nodes using 9 pin connectors Other nodes powered individually

Duplicate address detection .... Yes to some degree, Interbus-S compares the node type to what is expected at a specific physical location. There are no addresses to set. The master creates a virtual address during the initialization scans but this is based upon the physical position the bus.

Attendance check per scan .... Yes, token passing will stop when a node is missing.

Error detection .... Yes, CRC

Error correction .... No, erroneous data is flagged as bad and not used, nodes will wait until good data is sent on subsequent cycles

Address setting .... None, see duplicate address

Node parameter programming .... A protocol mode PCP, Peripheral Communications Protocol, can send a variety of information to the field nodes. PCP is optional and supported only when the field nodes have an additional microprocessor that supports PCP and when the network master supports PCP.

History .... This bus was designed by Phoenix Contact in the mid 1980s and released in 1987. Phoenix actively recruited partners starting in 1990 and opened the bus shortly after.

Summary: Interbus-S has a strong following by drive manufacturers. It is fast, deterministic, and efficient. It is designed to have a single host. The 16 bits per node limitation could be a problem for MMI, Man Machine Interface nodes. The remote nodes and the local nodes use the same chips but are implemented differently so these two types of nodes are not interchangeable.

The remote bus uses three different connectors, the 25 pin Sub D shell, 9 pin Sub D shell, and a 9 pin round connector. The 25 pin connector is used where the optional auto detect node feature requires 8 pair twisted wire and a drain shield wire. The transmission media of the local nodes requires 11 wires. A 15 pin sub D shell connector is used.

If a remote bus node dies or the remote bus is opened or shorted, the nodes physically before and after the fault will automatically reconfigure the physical network and continue to operate.

Profibus

Profibus is actually three buses. Profibus FMS (Fieldbus Message Specification), Profibus DP (Decentral Periphery), and Profibus PA (Process Automation). FMS is a higher level bus intended to operate with PLCs, PCs and higher level nodes. This is a token passing protocol with a network master. Profibus DP supports three masters. The masters then operate with field nodes as master-to-slaves. PA is an IS, Intrinsically Safe bus. The protocol is essentially the same as DP but the electrical/physical specifications are modified to satisfy the low voltage an current requirements on IS.

ProfibusDP reserves address 000 for the programming monitor and 001, 002, 003 for the three masters. It also reserves 126 and 127 for general broadcast messages. Therefore, all field nodes have addresses 126 and 127 in addition to their own unique address.

Profibus

Bus type................Token passing between a maximum of three masters. Master-to-slave, explicit and solicited messaging for field nodes.

Total number of nodes................126

Bus topology................Straight bus, although considerable branching can be accomplished with repeaters

Distance................9600 meters @ 9.6 KBps with 7 repeaters
9600 meters @ 19.2 KBps with 7 repeaters
9600 meters @ 93.75 KBps with 7 repeaters
8000 meters @ 187.5 KBps with 7 repeaters
4000 meters @ 500.0 KBps with 7 repeaters
500 meters @ 1.5 MBps with 4 repeaters
500 meters @ 12.0 MBps with 4 repeaters

Transmission media................2 wire twisted pair with a shield. Power for each node, microprocessor, inputs and outputs from separate unspecified source.

Transmission signal................Square wave digital signals, EIA RS-485 standard.

Input bits per node................2048 (allocated in bytes)

Output bits per node................2048 (allocated in bytes)

Speed................9.6 KBps
19.2 KBps
93.75 KBps
187.5 KBps
500.0 KBps
1.5 MBps
12.0 MBps requires advanced transceivers

Bus power................From separate source

Duplicate address detection................No

Attendance check per scan................Yes

Error detection................Yes, CRC

Error correction................Yes, repeat is requested

Address setting................Dip switch and software, Off line & On line

Node parameter programming................Yes

History................Profibus is Siemen's evolution of the ET 100 and ET 200 remote proprietary bus. It is a DIN standard and in the hands of two organizations, PNO and PTO.

Summary: These three buses are large and can literally handle half a million I/O. They fit well into the control pyramid paradigm.

Smart Distributed System SDS

Smart Distributed System (SDS) is based upon CANbus using the 11 bit arbitration standard. This 11 bit arbitration is broken down into 4 bits for the type (16 flavors) of messages and 7 bits for the MAC ID (128 addresses).

The variety of CANbus processors and software make this a viable bus. Over 40 million CANbus chips will be made in 1996, making CANbus base networks the most widely used non-office network. There are several hardware and software tools that have been developed for CANbus in the automotive industry that can be ported to SDS applications.

This digital signal bus has proven reliable even in the engine compartment on automobiles and trucks. The only negatives may be the distance and the limitations on branching. Because of the bitwise arbitration used during power-up and certain messaging types, this bus does not lend itself to repeaters. If in a large system, where both node count and distance are needed, SDS would make an ideal bus to possibly feed into another higher level bus. Significant branching could also be achieved with active hubs. This method would cost more but may be worth it when branching into areas where the physical mishaps could damage the bus power and data lines.

Smart Distributed System SDS

Bus type	CSMA/BA, explicit and unsolicited messaging, master exclusively writes to outputs. Explicit-solicited messaging is also used for initialization and attendance check.
.umber of nodes	128 addresses are available but the transmission media specification permits a maximum of 32 nodes at 1.0 Mbps and 64 nodes at 125, 250, and 500 KBps using specified cable. To make use of the maximum addresses available, detailed calculations must be made of all transmission media components on an application-specific basis.
Distance	1500 ft. trunkline, 12 feet branches @ 125 KBps 600 ft. trunkline, 6 feet branches @ 250 KBps 300 ft. trunkline, 3 feet branches @ 500 KBps 75 ft. trunkline, 1 foot branches @ 1.0 MBps
Transmission media	Trunkline – 2 wire twisted shielded cable with optional 2 wire bus power cable and drain wire.
Transmission signal	Square wave digital with NRZ (Non Return to Zero) encoding.
Input bits per node	32 bit model exists at this writing, potential for 64 exist, and fragment messages are supported
Output bits per node	32 bit model exists at this writing, potential for 64 exist, and fragmented messages are supported
Speed	125 KBps 250 KBps 500 KBps 1.0 MBps
Bus power	From trunkline
Duplicate address detection	Yes, the network master asks the nodes to announce their address and if node hears its address announced by another node it will not advance to a run mode.
Attendance check per scan	No. But most interfaces have the option to perform a polling type of attendance check at intervals.
Error detection	Yes, CRC
Error correction	Yes, nodes that detect errors signal the sender to repeat.
Address setting	Off line via hand held programmer or with a dedicated interface and host. On line via the interface master using a reserved newcomer default address that is changed to an application address. Dip/rotary switches are optional.
Node parameter programming	No standard, open to each vendor.
History	Smart Distributed System is from Microswitch-Honeywell. This bus was quietly introduced in 1993 and fully released at the 1994 ICEE show. Also at the show, Microswitch formally opened the bus to all reputable vendors.

Summary: This CANbus based network has been implemented with a more native approach than DeviceNet. It uses unsolicited messaging (interrupt driven) like the nodes that are on board vehicles and also has an attendance polling. When operated in interrupt driven mode, any input node can access the bus and talk when it detects a local change. Therefore, any interface node to a PLC or PC must listen, receive and process information at burst speed (maximum message rate) without dropping messages. When this is accomplished Smart Distributed System can be the most efficient network by reporting only changes. The user must decide if the loss of an attendance check on every scan is acceptable.

Seriplex

Seriplex received Control Engineering's product of the year award in 1990. It was designed for easy use; most anything to do with protocol was embedded in the ASIC and transparent to the user. It uses a system bus clock to pulse the nodes. The individual nodes listen to the clock line and when there have been enough binary pulses to equal the node's address, the node reports its information on the data line. This is how the input information is handled in either a limited peer-to-peer communication or master-to-slave.

There are two ways to handle the output information. Mode 1 is peer-to-peer and mode 2 is master-to-slave. The individual nodes are told their address and the mode type off line before installation in the network. Also, when programmed, the output nodes must be told whom to listen to, and the respective "ON/OFF" action to be taken. In the master-to-slave mode, the output nodes are written to only by the master. There are twice the number of clock pulses in the master-to-slave mode as there are in peer-to-peer mode. In the peer-to-peer, there is a one-to-one relationship between an input node and an output node. Nodes can be both input and output types.

Seriplex

Bus type	Serial using a bit shift method, soliciting messages. Outputs are written to by the master (mode 2) or by input (mode 1).
Total number of nodes	256
Bus topology	Free Form: Drops, Branches, Stars, Loops and Combinations
Distance	Using a 22 awg data and clock line at 16 pF/ft 500 feet @ 100 KHz Clock Speed 5,000 feet @ 16 KHz Clock Speed
Transmission media	5 Wire Cable Clock 22 awg 16pF/ft Data 22 awg 16pF/ft V+ 16 awg V_{com} 16 awg Drain Shield & Drain combination
Transmission signal	Digital
Input bits per node	1 (optional multiplex mode, see summary below)
Output bits per node	1 (optional multiplex mode, see summary below)
Speed	Read/write cycle for 1 bit per address. Master/Slave 64 Inputs and 64 Outputs @ 16 KHz = 8.50 msec Peer-to-peer 64 Inputs and 64 Outputs @ 16 KHz = 4.50 msec Master/Slave 64 Inputs and 64 Outputs @ 100 KHz = 1.36 msec Peer-to-peer 64 Inputs and 64 Outputs @ 100 KHz = 0.72 msec Second generation ASICs supports 200 KHz clock speed.
Bus power	From busline, see transmission media above
Duplicate address detection	No
Attendance check per scan	No
Error detection	The clock source generates a Bus Fault Detection pulse for detection of a faulty bus. The optional Data Echo provides data confirmation of output data. The optional digital debounce rejects spurious data.
Error correction	Continual data refresh provides data correction. Because the speed is so fast most inputs and outputs will be corrected by subsequent scans if there is an acutal error.
Address setting	Off line hand held setup tool
Address Parameter programming	Third output programming using Boolean logic. Non-Inversion and Inversion of all inputs and outputs available.
History	Seriplex Control Bus Products were originally developed and marketed by APC, Inc. Jacksonville, Mississippi. APC was purchased by Square D Company in 1995. The Seriplex Technology Oranization, formed in 1995, is a trade organization that actively recruites partners and promotes the technology since its inception.

Summary: The topology of the Seriplex Control Bus is free form making it an easy bus to install. The Seriplex ASIC has multiplex lines available to handle analog or larger amounts of discrete. Analog signals can be derived in either 8, 12 or 16 bit resolution. It is called multiplexed because only one bit of the resolution or discrete input points are transferred each scan. The Seriplex ASIC supports this multiplex operation in both directions (inputs and outputs). The Seriplex dual port RAM sets aside part of its RAM to manage the multiplexing.

Specialty buses

These buses are more closed than the general purpose buses. Both of these buses are pioneers, and if maturity (knowing where the bugs are) is important, then both of the buses could be good candidates.

Genius I/O

Genius I/O is one the original buses. It was introduced in 1985. The bus is robust and has superior data integrity. When messages are large, it can move a lot of data per node with a good data per overhead ratio. GE has implemented considerable diagnostics in the field I/O nodes. Today GE has positioned Genius as a bus to link small PLCs and as a mature (no bugs) device level network.

I almost placed this bus in the open general purpose buses, because GE has proven to be a willing partner with other vendors when a customer requests an integrated solution.

Genius I/O

Bus type	Token passing, solicited messaging, peer-to-peer
Total number of nodes	32
Bus topology	Straight bus with restricted branches
Distance	3500 ft @ 153.6 KBaud to 7500 ft @ 38.4 KBaud
Transmission media	2 wire twisted shielded cable with coax and fiber optic options
Transmission signal	Modulated using FSK, frequency shift key
Input bits per node	1024 bits (allocated in bytes)
Output bits per node	1024 bits (allocated in bytes)
Speed	38.4 KBaud to 153.6 KBaud (special 450 KBaud also available)
Bus power	Not specified as a Genius I/O standard; typically 24 VDC must be supplied or derived at each node
Duplicate address detection	Yes
Attendance check per scan	Yes
Error detection	Yes
Error correction	Yes
Address setting	Off line programmer by hand held or interface and host.
Node parameter programming	Yes

Summary: Genius I/O is robust and fairly fast when large amounts of data per node are being transmitted. The signal type with essentially on-the-fly corrections makes this the most deterministic bus on the market today with any maturity.

Sensoplex®

Sensoplex® is one of the original buses. It has been used for on-the-machine applications since 1987. The bus is robust and has excellent data integrity. The nodes are connected via a high-flex 75 Ohm coax cable. Both data and 24 VDC power for the inputs and node requirements flow on the coax. This bus uses a FSK modulated signal, (see appendix A for a detailed description of how this type of signal is used for error detection and duplicated address detection).

Sensoplex® IS (Intrinsically Safe) inputs and outputs comply with NAMUR standards of 8.2 VDC at 3 mA or less. The input and output stations may be located in the same area as the sensors and solenoids. Solenoids that comply with the NAMUR standards are now available from Samson, ASCO-Jouncomatic, and Hoerbinger. Sensoplex® IS is listed or approved by PTB/BVS of Germany for IEC/CENELEC standards, and CSA and FM of Canada and the United States for North American standards. IEC/CENELEC ratings are Class EEx ib IIB T4. CSA ratings are Class I, II, III, Division 1, Groups C, D, G. FM ratings are Class I, II, III, Division 1, Groups C, D, E, F, G.

Bus type	Sensoplex® Master-to-slave, first generation Sensoplex®2 Master-to-slave, second generation Sensoplex® IS Master-to-slave, Intrinsically Safe compatible with both first and second generation.
Total number of nodes	32 first generation, 64-120 second generation, depending upon host.
Bus topology	Straight feed-through bus, unrestricted branches.
Distance	200 meters between master and the most distant station, 400 meters between master and the most distant station with use of amplifier(s).
Transmission media	2 wire coax
Transmission signal	Phase continuous frequency shift key modulation
Input bits per node	8
Output bits per node	8
Speed	187 KBps
Bus power	Power and data on same coax
Duplicate address check	Yes, see appendix A for explanation
Attendance check per scan	Yes, see appendix A for explanation
Error detection	Yes, see appendix A for explanation
Error correction	Yes, master will request up to three times before declaring a fault
Address setting	Dip switches
Node parameter programming	No
History	This bus was originally developed for Ford Motor Co. in Köln (Cologne) Germany. The application required the nodes and data lines to be in direct contact with the robot welders. The situation required significant data integrity because of the close proximity of the magnetic field.

Summary: Sensoplex® has a proven record around resistive welders used in the manufacture of automobiles and trucks. These massive low frequency electro-magnetic fields affect not only the data, but the power to the nodes and attached inputs and outputs. The modulated signal using phase continuous FSK and filters maintains the data integrity. The coax, the input conditioning, and power regulation combine to overcome the affects of the fields on the power.

The Sensoplex® IS version uses the same master as the standard. The protocol is the same as are other passive components. This version has been approved for use in Europe and North America.

Light Buses

Light buses use light signals over glass or plastic fiber optic cable. A light emitting diode is driven on and off to form a binary signal by the sender. The listener has a light sensitive transistor that converts the light signal to an electrical signal used by the node's chip sets. Most buses can be adapted to light with a special transceiver, but usually something is lost or complicated in the adaptation.

When designing a bus, the main difference between electrical and light transmission is the ability to branch from the main bus line. An electrical voltage signal is easy to branch. Even in the most restrictive situations a half a meter is possible. But it is very difficult to branch light on a fiber optic cable. It is done in some factories but almost never in the field. Therefore the best way to make a light signal bus is to use fast components and make every node a repeater. Since every node is a repeater this lends itself to a physical ring, but not CSMA buses.

CSMA/BA buses are best used on a single segment electrical bus. To perform arbitration (see chapter 5) all the nodes must be hearing the same bit at the same time. In theory the same bit must cover the entire bus at the same time. One bit is on for 2000 ns at a speed of 500 KBps. A data signal will travel down a 100 meter data line in about 750 ns. So, only 1/8 (750/2000) of a bit is on the data line at one time. In other words, all nodes are within 1/8 of a bit of one another. A repeater using ultra-fast components, comprising of a transciever-optocoupler-transceiver has a through put time delay of about 180 ns. If we had five repeaters on this bus, we would have 900 ns of time delay, therefore our bus would have to be a negative length to acheive a 750 ns end-to-end signal time.

The CSMA/CD buses are also hindered by repeaters. CD (Collision Detection) works well on a single segment bus because if more than one node sends a message at the same time, the senders hear the collision at almost the same time. The senders recognize the collision and after a random time delay, repeat their message. On the other hand, a bus with 100 repeater nodes won't work because the end nodes are essentially 50 bits different. Messages sent by nodes on both ends could collide in the center, well after the nodes had finished sending what they thought was a sucessful transmission.

The result is that CSMA type buses are poor candidates for light buses. Either a master-to-slave or token passing type buses are typically used. Token passing is the most efficient.

The disadvantage of a light signal bus is the requirement for power. Two copper wires used with an electrical bus can provide data and power for the local node and sensors as in the case of ASI and Sensoplex®. But a bus using a light signal must have another power source. This is not a problem if the nodes are drives, man-machine-interface panels, or a high-level instrument – DC power is usually available. But for sensors and simple limit switches, power must be provided by a fused electrical source.

The advantages of a light bus are speed and inherent noise immunity. This is good news for drive manufacturers who also want a deterministic bus. There are a few disadvantages, such as the increased cost of the transmission media (fiber optic cable) and the cost for a light signal transceiver. The cost of the transceiver is nothing when compared to a drive but significant if used on a $25.00 limit switch.

These buses are a good choice for drive positioning and coordination. These buses are overkill for discrete sensors, limit switches or pushbuttons. So either run the discrete point directly back to a PLC input, or use high density

discrete input nodes. By high density, I mean at least 4 point if not 8, 16 or 32.

The two light buses I will talk about are the **Beckhoff Light Bus** and **Sercos**. Both are European. Beckhoff is said to be the largest manufacturer of industrial computers in Germany. The light bus they have designed is mostly closed. The transceiver and ASIC are of Beckhoff design and must be bought from Beckoff.

Beckhoff Light Bus

Bus type	Token passing, explicit and solicited messaging. Master exclusively writes to field nodes.
Total number of nodes	256
Bus topology	Ring type with repeater nodes.
Distance	45 meters between repeater nodes with plastic fiber. 600 meters between repeater nodes with glass fiber.
Transmission media	Glass and plastic fiber optics.
Transmission signal	Binary light
Input bits per node	32
Output bits per node	32
Speed	2.5 MBbps
Bus power	Separate supply source
Duplicate address check	Yes, because each node is also a repeater, two nodes with the same address would have an out-of-synch (detectable) data collision.
Attendance check per scan	Yes, master check list
Error detection	CRC
Error correction	Yes, repeat will be requested
Address setting	Dip switches
Node parameter programming	Yes

Summary: There are interfaces to Siemens PLCs. Most of Beckhoff interface products are Intel based coprocessor boards for use with DOS or Windows.

Sercos

Bus type	Token passing, explicit and solicited messaging. master exclusively writes to output field nodes.
Total number of nodes	256
Bus topology	Ring type with repeater nodes.
Distance	60 meters between repeater nodes with plastic fiber 250 meters between repeater nodes with glass fiber.
Transmission media	Glass and plastic fiber optics.
Transmission signal	Binary light
Input bits per node	32
Output bits per node	32
Speed	2 MBbps
Bus power	Separate supply source
Duplicate address check	Yes, because each node is also a repeater, two nodes with the same address would have an out-of-synch (detectable) data collision.
Attendance check per scan	Yes, master check list
Error detection	CRC
Error correction	Yes, repeat will be requested
Address setting	Dip switches
Node parameter programming	Yes

Summary: Sercos is fairly open, although the software license for a Sercos compliant interface is $7,400. Considerable development products are available.

Chapter 7

Common Relief For A Sick Bus

This chapter can't attempt to cover every bus malady. But once a bus has worked properly, there are still some common ailments that can come up from time-to-time. These ailments that I will address are sort of like a head-cold— under ideal conditions they are barely noticeable, but if the system is stretched to the limits (distance, speed or the node count), a collapse could happen.

CPU problems are for the brain specialist

I don't believe in mail order brain surgery diplomas so we will leave the CPU surgery for the specialist.

When the CPUs or ASICs in the nodes aren't properly executing their program, anything can happen. The anything is usually nothing. The program may execute to a specific point, then crash, then nothing. Fortunately, this type of fatal error is found while still in development, under the care of a specialist.

The second kind of brain problem happens in the alpha and beta stages of a new product. Printed circuit boards at these stage are partially hand made. Some of the CPUs and ASICs have 60 or more leads about a millimeter apart that are a little larger than a few strands of human hair. Most soldering errors are caught at point of manufacturing, but a few weak joints could pass tests. The vibration caused by shipping or on-the-machine usage could break the weak joints. Very little can be done in the field when these problems occur. My advice is, "When buying any pre-released industrial electronic product, have lots of spares. And put it through considerable testing before putting your company's business in jeopardy." Most manufacturers will be more than happy to consign extra spares on their own money to have a test site. But they will probably want a disclaimer signed, a mutual acknowledgment that the products is pre-released and the buyer is applying them at his own risk.

If these alpha and beta products sound too risky, then wait until the product has been fully released. Then still buy a spare or two.

Data line connections

Wiring problems are by far the most common. Good clean data line connections are mandatory. It is like having good clean oil in your car engine. If you don't, it is just a matter of time until you will have a massive problem.

Poor connections of data line are cumulative. Ideally, connections that are exposed to the environment should be plated with a conductive non-reactive metal. Gold is the best. It is used on cards that fit into our PCs and PLCs and on telephone jacks.

Most manufacturers of the industrial data cable products use gold plated pin and sleeve construction for connectors (see figure 8-1). If a connector is subject to a number of insertions and removals, or subject to abnormal alignment, the tension in the sleeve should be spring reinforced. The cleaned and tinned wires are either soldered or crimped to the pins and sleeves in a clean factory environment. Then a rubberized plastic is molded around the solder or crimp joint that seals them from the environment and restrains mechanical stress.

Industrial busline connectors
Figure 7-1

Not all connections are possible with factory made cables with molded connectors. There are some lengths that cannot be determined until those hectic hours before an installation is commissioned. It may be the run from the first node to the controller, or it may be between sections of a machine. My experience has been that over 90% of the on-the-machine wiring can be done with pre-made connectorized cables.

The data line runs that are inside cabinets such as motor control centers or control panels are more forgiving than field wiring. Once the start-up is done and the door closed, the connectors and cable are not usually flexed and stressed. The length of the runs inside the enclosures is shorter – again more forgiving.

Whether in the cabinet or on-the-machine, if screw terminal connections must be made, there are good ways and there are bad ways.

Bad ways:

- Use corroded or dirty screw terminals
- Use oversized screw terminals
- Jam too many wires under the screw terminals
- Screw down on the insulation not the wires
- Nick the insulation

Good ways:

- Use gold, silver, or tin plated screw terminals
- Tin all bare copper wire
- Tin pre-tinned wire if any strands tend to fray
- Use a vapor tight box and corrosion resistant compound in corrosive or wet locations
- In this order of importance use:
  Specified screw terminals
  Screw terminals designed for data lines
  Screw terminals with teeth
- Provide strain relief for any cable that is subject to flexing or stressing

The importance of the shield

Shielding is a method to *reflect* or *drain* harmful signals also called interference. Shields can be used around conductors, inside or outside of an enclosure or around an entire room.

A common example of a reflecting shield is the window of a microwave oven. A metal screen is layered-on or embedded in the glass window. The screen becomes a mirror for these short wavelengths. In fact, wavelengths from 0.1 meter (radar) to 10 Angstrom (end of ultra violet range) can be shielded in a reflective manner. Wavelength shorter than 10 Angstrom will have no direct effect on light or electrical signals. Metal foil is an example of a good reflective shield.

The wavelengths 0.1 meter to 10,000,000 meters are the problem. This is an electro-magnetic field radiated from a device such as a radar to a standard power transformer.

The interference from wavelengths of 0.5 meter to 10.0 meters has in recent years been the biggest problem. Notice, I didn't call it noise, because although it may be noise to the bus it is not noise to the intended signal. In this range, there are many types of radio signals. Broadcasts from commercial sources are

usually not a problem unless you are extremely close to the transmitter. The energy from the source dissipates inversely proportional to the distance cubed. The real problem is from mobile sources such as walkie-talkies.

An aluminum foil with drain wire is an effective solution against these wavelengths. The drain wire must be terminated to ground at only one point. The theorist will say the termination point should be at a mid-point on the bus. The practical person will say at the source. This usually is in the enclosure where the master node, PLC and bus power are located. The common and overwhelming argument is that the ground point in the enclosure is easier to establish and maintain than a point on the machine.

Most vendors of industrial electronics had learned to live with 3-watt walkie-talkies in the 1980s. About 1990, the first 5-watt walkie-talkies started to appear (only 4 watts worth of energy was needed to communicate from Earth to the moon during the Apollo program). These 5-watt walkie-talkies brought down regulators, switching power supplies, altered capacitor type oscillators, and triggered transistors. **This happened on the power circuits as well as the signal circuits!** Those who believe a light bus frees the system of electro-magnetic interference are wrong. The power circuits in an electronic device are just as important as the signal circuits – if one is down, the other doesn't matter. The vendors have learned to live with 5-watt walkie-talkies but it is at the practical limit of hardening a circuit board via intelligent design. Any more hardening for higher wattage will require additional and costlier components – that expense will in one way or another come out of the user's pocket.

As the wavelengths become longer, the usefulness of aluminum foil shielding diminishes. Most industrial buses either use signal techniques and/or twisted data lines with braided shield to nullify the effect of longer wavelengths (see chapter 3 - topic The Signal). This is a science in the labs, but on the factory floor or process plant, multiple sources of interference can combine to push this science to an art form.

Fortunately, there is a back up if your bus becomes bogged down with repeats. Look for the source of interference. Welders and SCR controlled equipment are a likely source. Also, equipment with power MOSFETs or IGTOs (new power conditioner and drive power components) could be a problem source. Other sources could be anything that makes an arc or uses electro-magnetic radiation in the process, such as RF furnaces. Start with the equipment having the highest wattage. Turn off the power to the equipment, and run the bus. Does the performance of the bus impove? If not, try the potential source with the next highest wattage. If the source can be identified, then run the bus data and power lines inside ferrous conduit for 5 meters before and after the source. Ground the conduit to any available stable ground; this ground does not have to be the high quality ground talked about before in reference to the drain wire of an aluminum shield.

What can you do with common tools?

If portions of the bus are dead, the bus bogs down with repeats, the nodes are dropping out, or changing the actual location of one or more nodes (within specification) causes miracle cures or mystifying problems, then we are usually dealing with physical problems. **These are wire problems, connector problems, transceiver problems, layout problems. These are not message, software, CPU, or protocol problems.** Many of these problems can be found and resolved with good a digital VOM meter.

In the next few troubleshooting tips, I've assumed we have minimal diagnostic indication from the nodes, whether LEDs or messages. If you chose a bus and nodes with extensive communication diagnostics, many of these tips are superfluous. You would use them only when the diagnostic indicators can't tell the whole story.

Data line problems–grounded, shorted, open, or flip-flopped

Shorted - open data line or shorted transceiver

While this may not be the most common problem, it is easy to check. First isolate the two data lines by turning off all power to any nodes. Remove any termination resistors and leave all the nodes in place and connected. Check the resistance between the two data lines. The resistance is the parallel summation of all the transceivers given by $1/R^{Total} = 1/R^{transceiver\ 1} + 1/R^{transceiver\ 2} + \ldots 1/R^{transceiver\ n}$. A CANbus transceiver has a minimum differential resistance of 20 K Ohms and a maximum of 100 K Ohms. Therefore, the following is a chart for expected resistance of a bus with a certain number of nodes:

Nodes	K Ohms Min.-Max.	Nodes	K Ohms Min.-Max.
1	20.00-100.00	19	1.05-5.26
2	10.00-50.00	20	1.00-5.00
3	6.70-33.00	21	0.95-4.76
4	5.00-25.00	22	0.91-4.55
5	4.00-20.00	23	0.87-4.35
6	3.30-16.67	24	0.83-417
7	2.90-14.29	25	0.80-4.00
8	2.50-12.50	26	0.77-3.85
9	2.20-11.10	27	0.74-3.70
10	2.00-10.00	28	0.71-3.58
11	1.82-9.10	29	0.69-3.45
12	1.67-8.33	30	0.67-3.33
13	1.54-7.70	31	0.65-3.23
14	1.43-7.14	32	0.63-3.12
15	1.33-6.67	33	0.60-3.03
16	1.25-6.25	34	0.59-2.94
17	1.88-5.88	35	0.57-2.86
18	1.11-5.55	36	0.55-2.78

Nodes	K Ohms Min.-Max.	Nodes	K Ohms Min.-Max.
37	0.54-2.70	51	0.39-1.96
38	0.53-2.63	52	0.38-1.92
39	0.51-2.56	53	0.38-1.89
40	0.50-2.50	54	0.37-1.85
41	0.49-2.44	55	0.36-1.82
42	0.48-2.38	56	0.36-1.79
43	0.47-2.33	57	0.35-1.75
44	0.45-2.27	58	0.34-1.72
45	0.44-2.22	59	0.34-1.69
46	0.43-2.17	60	0.33-1.67
47	0.43-2.13	61	0.33-1.64
48	0.42-2.08	62	0.32-1.61
49	0.41-2.04	63	0.32-1.59
50	0.40-2.00	64	0.31-1.5

EIA RS-485 type transceivers have a 12 K Ohm minimum value. I could not get a stated or published maximum from either of two major transceiver manufacturers. The EIA RS-485 specifies a maximum of 32 unit loads (transceivers) on a segment.

Nodes	K Ohms Min.	Nodes	K Ohms Min.
1	12.00	17	0.71
2	6.00	18	0.67
3	4.00	19	0.63
4	3.00	20	0.60
5	2.40	21	0.57
6	2.00	22	0.55
7	1.71	23	0.52
8	1.50	24	0.50
9	1.33	25	0.48
10	1.20	26	0.46
11	1.09	27	0.44
12	1.00	28	0.43
13	0.92	29	0.41
14	0.86	30	0.40
15	0.80	31	0.39
16	0.75	32	0.38

If the resistance is less than the above values, there is probably a short somewhere. Divide the system in half, test again, and check the values. Continue to halve, test, and check the part of system that is less than the preceding values until the problem is found.

An open data line is more difficult to find. The preceding values may give you a hint, but the minimum and maximum vary so much that an open condition is not conclussive. If you suspect an open, measure the bus at the beginning and at the end. The resistance should be the same. If not, halve the system, check, and test until the open is found.

Grounded data lines

To check for a grounded data line, again turn off all power to the nodes. Disconnect any terminating resistors. Check the resistance to ground on each data line. It should be nearly infinite. Reverse the meter polarity in case the lines are protected by diodes. If the resistance is not nearly infinite, then divide the system in half and test again. As before, continue to halve the system until the ground is found.

Flip-flopped data lines

There is one more check that can be done on the data lines. It is very possible to flip-flop the two data lines during installation, add-ons, or a physical relocation. If this happens, there will usually be a large portion of the bus that is dead. Disconnect the nodes from any power source. Disconnect any terminating resistors. At the end of the bus, jumper one of the data lines to the drain wire (you should have already performed the shield test described in the next topic). Test this data line–drain wire circuit back in the cabinet at the start of the bus. It should read nearly zero resistance. If it does, test the other data line, after re-configuring the jumper from the drain wire to the other data line. The resistance should be nearly zero. If either test results in some resistance, then the data lines are probably flip-flopped. Unfortunately, this test can't find two flip-flops. If I suspect a flip-flop or multiple flip-flops, I prefer to break the bus into sections (maybe 2-3 nodes per section), then install the terminating resistor at the end of the first section. Finally power-up and check to see if the nodes are communicating. Keep adding sections until the fault is found.

This is the hard way to look for a flip-flop. Today, most nodes feature good communication diagnostics and indications. You could also find the problem by just walking down the bus from the beginning, and then looking at the LEDs for the first node that wasn't communicating. Then check the connection just before the node.

While you are checking the data lines, check the terminating resistors, if they are used. CANbus and EIA RS-485 use 120 Ohm terminating resistors at the beginning and end. Coax has various resistances, so read the specs.

A grounded or open shield

The shield is important and either a ground or open can be one of those nasty problems that has ambiguous symptoms. Fortunately, this is easy to check in just seconds. Find the one point where the shield should be grounded, usually back at the PLC. Power down, and disconnect the shield drain wire from the ground. Then check the resistance from the drain wire to ground. The resistance should be nearly infinite. If not, then divide the system in half and test again. Continue to halve and test the system until the ground is found. Remember to reconnect the single point ground when finished.

Voltage and current checks of the bus power lines

No matter what bus we are talking about, the brains, CPU or ASIC must get power. Peripheral components also need power. Even momentary drops below

the specified voltage cause the nodes to fault and then go through the start-up procedure.

Checking the voltage is not difficult. Install some sort of Tee on the power bus at a point farthest from the power source. Check the bus power specifications. Verify which lines are the power. Then use a digital VOM meter and check the voltage while the bus is operational, if possible.

If the voltage is low, then either:

- The power source is inadequate for the load.
- One or more high resistance connections are in the system.
- There is too much cumulative resistance in the wire and connections.

Go back to the power source. Check the current and voltage. If both the voltage and current are less than the rated output, replace the power supply. The rated output may or may not be sufficient to handle the load but this power supply has a problem.

If the current is at or above the rated output and the voltage is at or below the rated output, (measured at the power supply), then you need more power. Consider a larger power supply or multiple power supplies. There are only a couple of right ways to do multiple power supplies and many wrong ways. If you don't have the knowledge, consult with someone that knows the bus and power supplies.

If the voltage is at the rated output and the current is below the rated output, (measured at the power supply), then there is too much resistance in the circuit. The hard part comes in determining if the voltage is a gradual accumulation of resistance or just one or two bad connections. Start again using the Tee at a point farthest from the power source. Make voltage checks at each connection, working your way back to the source. Record the voltage. If there are a few bad connections, the voltage drop should stand out. If so, fix or replace the hardware at the bad connections.

If the voltage drop is gradually cumulative, use the same power source to find a way to feed the power bus at multiple points. If it is a straight line bus, try powering it at both ends.

Custom diagnostic tools

This tutorial can't go into all of the possible signal, data line, and message analyzers available. Each bus is different. If this is important to you, I suggest you talk to the field service specialists of leading vendors. Ask about the merits of these tools and the learning curve to master their usage.

Appendix A

Error and duplicate address detection when using Phase Continuous Frequency Shift Key Modulated Signal

Duplicate address detection

Most buses in some way or another have duplicate address detection. This is a fundamental necessity for machine tool operation. Just the possibility of a node at the end of a conveyor answering for a node in the guts of punch press would captivate the interest of many personal injury lawyers.

Buses that use a digital square wave signal must rely upon methods that use either a physical order check or an initialization address announcement. True ring buses, where every node is a repeater, have several options all based on the physical order. CSMA/xx buses use methods based upon address announcement during power-up.

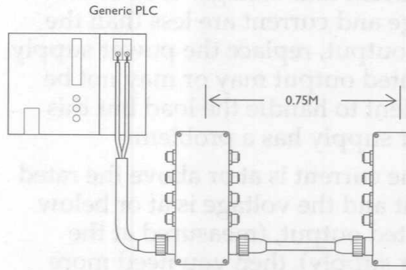
Buses using modulated signals have another possible method for detecting a duplicate address. This method is based upon the length of cable between adjacent nodes.

Almost the speed of light

In common usage around the home and office electricity is nearly instantaneous. There is a time delay from the moment a switch makes contact to the moment an incandescent light becomes bright. Most of the time delay is due to the heating of the incandescent filament, but there is a minuscule delay due to the completion of the circuit. For humans, who have a visual perception speed of about 20 Hertz, this delay is impossible to discern. But for a data transmission, the velocity (distance/time) of an electrical signal traveling on a wire is a determining parameter.

The terminology used to describe the velocity of an electrical signal is called the propagation speed, and is given in a percentage. A typical minimum prorogation speed for a twisted pair of data quality wires is 72%. This means a signal will travel the wires at 72% of the speed of light in a vacuum. Coax is a little faster, so for this example (because the math is easier) we will say 80%.

Figure A-1 shows a portion of a master-to-slave bus. This is a simple master-to-slave, where the master controls network management, timing, and communications. The two field nodes, slaves, are 0.75 meters apart with the same address of 05. Someone made a simple mistake, they should have been 05 and 15. Both nodes are identical and until the machine goes into a run mode both nodes have identical data to report.



PLC, one piece master station/gateway, and two field nodes (slaves)
Figure A-1

At some time during the first scan the master sends an explicit message telling node 05 to report. The message travels down the coax and is heard by the first physical 05 node and minuscule time delay later by the second 05 node. The first 05 starts talking earliest because it heard the request message first, (figure A-2a). The second 05 node reports slightly later, (figure A-2b) because of the time it took for the message to travel the extra 0.75 meters. Figure A-2c shows the first bit of the combined signals of the two 05 nodes just before it enters the transceiver of the master station. The identical information from

the second 05 node is now delayed by the time it took the signal to go 1.5 meters. Why 1.5 meters, not 0.75 meters? Because the original request from the master had 0.75 meters more distance to get to the second node and the report from the second 05 node has 0.75 meters more distance to get back to the master. Therefore, 0.75 meters + 0.75 meters = 1.5 meters.

Figure A-2d represents the combined signal with respect to the transceiver. I won't go into the decoding of this signal, but the result would obviously be junk. The master's CPU will consider this message an error and depending upon its program, it may try address 05 a few more times before declaring a fault.

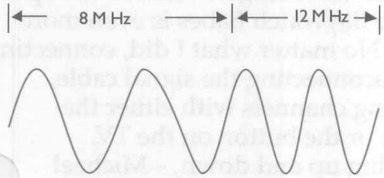
Drawing 4 parts

a - signal from first node

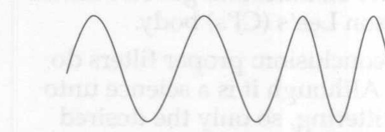
b - signal from second node

c - superimposed signal

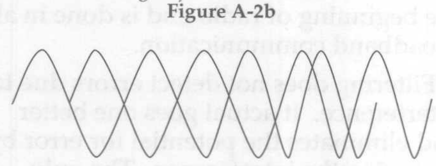
d - resulting signal



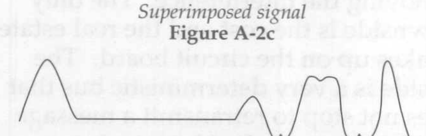
Signal from first node
Figure A-2a



Signal from second node
Figure A-2b



Superimposed signal
Figure A-2c



Resulting Signal
Figure A-2d

The math:

- Propagation speed 80%.
- Speed of light in a vacuum 3×10^8 meters/second.
- Frequency of the lowest the signal (in this example, we are using a common 8 megahertz and 12 megahertz for the signals and therefore, 8 megahertz is used in the calculations).
- Total distance of 1.5 meters
- Signal speed on the coax = propagation speed x the speed of light

  $= 0.80 \times (3 \times 10^8)$

  $= 2.40 \times 10^8$ meters/second
- Time for one cycle at 8 megahertz

  $= \dfrac{1}{8 \times 10^6 \text{ cycles/seconds}}$

  $= 1.25 \times 10^{-7}$ seconds
- Time for one half cycle, to accomplish a 180 degree phase shift as shown in Figure A-2 b and c

  $= \dfrac{1.25 \times 10^{-7} \text{ seconds}}{2}$

  $= 6.25 \times 10^{-8}$ seconds
- Distance transitioned in 6.25×10^{-8} seconds

  $= (2.40 \times 10^8 \text{ m/sec}) \times (6.25 \times 10^{-8} \text{ sec})$

  $= 1.5$ meters
- Separation of the nodes

  $= \dfrac{1.5 \text{ meters}}{2}$

  $= 0.75$ meters

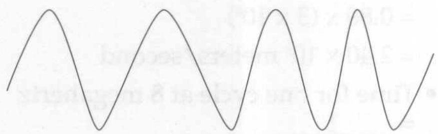
So, if the nodes in this example are 0.75 meters apart, the combined transmission back to the master is totally useless — this is what you want. The smart master will declare a fault because of a bad signal from node 05 and then idle the bus until the error is corrected.

If the bus you have uses this method, check the specs. Typically, the engineers will specify 10-30% more distance just to make sure they cover any variance in

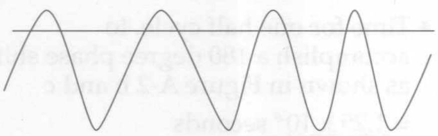
wire or the oscillators (clocks) in the nodes.

Error caused by signal interference

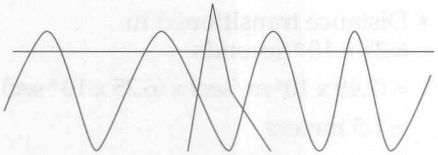
As discussed in chapter 3, the modulated signal is the most immune to external interference if proper filters are used. These filters eliminate all extraneous signals except for the frequencies that carry the data. Figure A-3 a shows a pristine 8 megahertz - 12 megahertz sine wave. Figures A-3 b,c, and d show interference from different sources. The last illustration, A-3 e, shows the signal after being filtered.



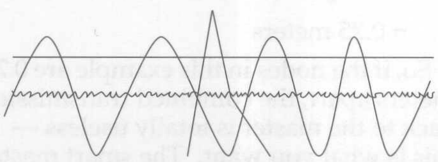
Modulated signal
Figure A-3a



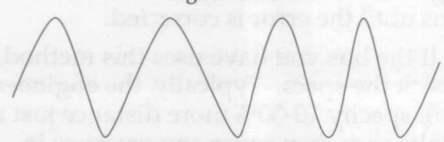
Interference from a low frequency source added
Figure A-3b



Interference from a solid-state switch source; SCR, transistor, etc.
Figure A-3c



Interference from a high frequency source such as a walkie-talkie
Figure A-3d



Modulated signal with interference but after filtering
Figure A-3e

Practical real world filters

To tell you the truth, I've run out of techie talk – the well is dry. Even though there are two more appendixe in this tutorial, I've actually already written them. This is the last topic for me, so how do I describe filters without getting into more technical words? How about a practical real world experiment!

On my home television, as I write this last topic, are reruns of "Baywatch" and "Star Trek, The Next Generation". Each one is on a different channel. I've checked over the TV and there is only one cable for the signals. So both programs must be on the same wires.

I tried to imagine a gross industrial bus incident and equate it to my experiment. Interference that would cause a reversing starter to change state and reverse the direction of a motor is pretty gross. Interference between the two signals that would result in putting the head of a Klingon Warrior on top of one the Baywatch Babes is even more gross. No matter what I did, connecting and disconnecting the signal cable, changing channels with either the remote or the button on the TV, powering up and down, – Michael Dorn's (Worf's) head did not go over the other channel and get on Pamela Anderson Lee's (CJ's) body.

The conclusion: proper filters do work. Although it is a science unto itself, filtering, so only the desired signal is obtained, has been done since the beginning of radio and is done in all broadband communication.

Filtering does not detect errors due to interference. It actual goes one better and eliminates the potential for error by removing the interference. The only downside is the cost and the real estate it takes up on the circuit board. The upside is a very deterministic bus that does not stop to retransmit a message that has been trashed by interference.

Appendix B

ISO-OSI Seven Layer Network Model

When is the Model Important?

It seems that whenever a group of people get together to talk about data communication, an overhead of the OSI Seven Layer Data Communication Model must be shown to sanction the meeting. Networks served us well long before this model was created. But in defense of the writers, it is a model that breaks down data communications so that no matter how dissimilar the networks actually are, we can define the transmission of data from a sender to a receiver.

Maybe because it is so comprehensive, there is a tendency to think of it as a functional checklist that correlates to the value of the network. This may be true of the enterprise level networks (I am not qualified to answer that), but it is not true at the device level. The network that is fast, simple to use, and works at two in the morning is usually the best network.

Dissecting Data Communication

Gateways, field nodes, and master type nodes will usually support in some manner all seven layers. In the case of a simple master-to-slave bus, many of the layers and the functions performed are trivialized. It is like having only one name on a ballot – there is no other choice. So rules are not embedded in software or firmware to handle the non-existent alternative. The field node in a simple master-to-slave only using layers 1, 2, and 7 (see figure B-1). A token passing ring bus only uses 1, 2 and 7 during normal run operation. During a fault, such as a cut data line, the nodes adjacent to the cut will use the routing functions in layer 3 to reconfigure the bus.

So, what do the layers in figure B-1 mean? Before we go on, I should note that this is not the literal ISO-OSI definition. I've synthesized parts of it to make it more relevant to device level buses and condensed it to fit in this small appendix. If there are errors it is my fault.

I wanted the examples to be universal, but they were too vague. So I wrote this thinking about a PLC host with a gateway interface card in a PLC slot, operating as a master-to-slave bus, using only solicited messaging, inputs report only to the master, and outputs only are written to by the master.

Example of Functions

Controller/host	Gateway (Interface module)	Input/Output Nodes	Layer	Function in OSI Model
PLC user program			*User program*	*not part of OSI model*
	Duplicate address detection and some types of error correction Presenting to host input data from field nodes	Duplicate address detection and some types of error correction. Presenting output data from the host to the local processor for action	Layer 7 Program/ application	Provides OSI services directly comprehensible to application program.
			Layer 6 Presentation	Restructures data to/from standardized format used within the network.
			Layer 5 Session	Management and control of data flow. Synchronizes and manages activity.
			Layer 4 Transport	Provides transparent reliable data transfer from all nodes.
			Layer 3 Network	Performs inter and infra routing of message packets. Any time an address is read or written to send a message this layer used.
	Accessing the bus, CRC check, divide and join fragmented messages	Accessing the bus, CRC check, divide and join fragmented messages.	Layer 2	Method to access the network. Rules to break-up and reassemble large blocks of data. Bit level error detection and correction.
	Bit level encoding, sending the signal (digital, modulated, light).	Bit level encoding, sending the signal (digital, modulated, light).	Layer 1 Physical	Encodes and physically transfers messages between nodes.
	Bus topology; electrical or light values of the signal; wire or fiber optics specs; connectors specs.	*Bus topology; electrical or light values of the signal, wire or fiber optics specs; connectors specs.*	*Physical Link*	*Not part of OSI.*

Figure B-1

Appendix C

Glossary

Active hub - a multiple port repeater or amplifier that lengthens the branching ability of a bus.

Address - a settable identifier for nodes. Also called the MAC ID (*Media Access* Identification).

Amplifier - a product that amplifies (strengthens) a signal in real time but precisely copies the old signal. This product links two portions of the same bus together. An amplifier is used when the signal is weak but not distorted. A human analogy is the use of a bull horn to communicate with a person several hundred meters away. (See *repeater* — a similar product)

Binary logic - a parameter used to describe a signal. A transmission using *binary logic* has only two states, "On" and "Off". The "On" usually means "High". The "Off" means "Low" or no signal. The information therefore must be structured and timed so that the "Off" is understood.

Bit - one binary piece of data that means either "High or Low," "1 or 0," "On or Off" or "Dominate or Recessive"

Bit encoding - puts a time reference to an electrical or electro-magnetic signal. This reference then creates a time based window so that the signal within the window can be evaluated and a determination made to the state of the bit, i.e. "High" or "Low".

Bitwise arbitration - See *message collision*.

Branch - a bus topology term used to describe a drop off the trunkline.

Bridge - a smart *repeater* that only repeats the data between two bus segments when the source and destination address are in different segments. The bridge must be programmed to know what addresses are on the respective segments. There is a several bit wait state as the bridge reads the address in the message header. The bridge may be a possible product for a device level bus but not common today. The bridge does not have an address.

Bus topology - the physical layout of the nodes and the interconnecting physical media.

Busline - any type of wires that carry data from node-to-node.

Byte - 8 bits of information.

Carrier - an electrical or electro-magnetic transmission that does not represent data but is used as a messenger on which the data is transported.

Collision detection - See *message collision*.

CSMA media access - an access method that allows each node a chance to transmit providing it has something to transmit, and only when other nodes are not transmitting. The CSMA is an acronym for Carrier Sense Media Access. If two or more nodes transmit at the same instant, some portion of the messages will collide. See *message collision*.

Datagram - a more techie word for message, but more open to a customized definition.

Device level bus - A industrial bus that connects basic control elements together or to a host controller.

Digital signal - a signal where a single change of energy level represents a single bit change. An example is a change of voltage from -1 volt to + 1 volt represents a change from "Low" to "High".

Dominant bit - CANbus uses a *dominant* and *recessive* to describe the binary state of a signal.

Dropline - a branch from a trunkline, usually of smaller size than the trunkline.

EIA RS-485 - a differential signal that defines the number of signal generators, receivers, or a combination of the two called a transceiver. It also defines the electrical values. It does not define anything about the message or the connecting wire.

End of message - a certain grouping of bits that indicates a message is over.

Explicit messaging - a command or order.

Frame size - bits used within a message to indicate the size in bits or bytes of specific data that will follow.

FSK - Frequency Shift Key - a common modulated signals method.

Gateway - a special node on two different buses that serves as a signal and data translator between the buses.

Interface card, interface module - generic terms for the gateway either in a PLC or PC that interfaces the host's bus to a device level bus.

Manchester - a common bit encoding method of digital signals that looks at the center of a bit for the "High" or "Low" state.

Master control media access- one super node controls all transmission, sequence and time. The remainder nodes don't talk unless told to by the master.

Media access - this is the "right-of-way" for transmission of data on a bus by the nodes. There are three main types. See *master control media access, token passing media access*, and *CSMA media access*.

Message - one complete group of continuous bits from beginning to end.

Message collision - when two people talk simultaneously there is a collision. A few humans can talk and listen to someone else at the same time but nodes can't. Both token passing and master control do not have that problem. A field node can only access the bus when it is its turn to talk or when it has been told to talk. The CSMA type buses do have this problem. Sometimes they are operated as a master-to-slave, so the problem goes away. When operating in the native CSMA access method, two nodes could start talking simultaneously. So two major ways have been developed to handle this:

BA - Bitwise arbitration - all nodes must be both senders and receivers. The busline must be of a specific length or less so that all nodes hear the bit at the same time.

CD - collision detection - each node must be both a sender and a receiver. If two nodes start talking at the same time, they will hear a collision. Both stop talking, wait a

random length of time, then look for a clear line to start talking again.

Modulated signal - a signal that uses two electrical or electro-magnetic frequencies to represent the two bit states, i.e. "High or Low". The most common and useful modulated signal is the sine wave signal.

Multiplex - a method to transmit numerous messages in sequence over two wires.

Node - an addressable device on the bus.

Non-destructive arbitration - See *message collision*.

Non-return to zero level - Also abbreviated NRZ-L. This bit encoding method is used with differential signals such as CANbus. With this method "0" is the "High" level and "1" is the low level.

Passive hub - multiple port Tee.

Physical media - the wire or optical cable that is used to transmit the data from node to node. Usually connectors and the components that transmit or receive the signal are considered physical media.

Protocol - a small program that is embedded in nodes to organize, decipher and react to the transmitted data.

Receiver - a electronic component that is coupled to the physical media. It is the ears to central processing unit or other logic components.

Recessive - CANbus uses a *dominant* and *recessive* to describe the binary state of a signal.

Repeater - a product that reconstructs a signal but with an associated wait state. The wait state is dependent upon the decoding and encoding method. It links two portions of the same bus together. A repeater is used when a signal is weak or distorted. A human analogy is to communicate a message to one person then have him/her repeat the message to another until it reaches the destination. (See *amplifier* — a similar product)

Ring - a term used to describe a bus topology where the trunkline is an unbroken loop.

Router - a higher level *bridge* for connection of wide area networks. This product would seldom be used on a device level bus. The destination network and destination address are included in the header of the message.

Scanner module - Allen-Bradley definition of the gateway product that plugs into their PLC and interfaces the PLC's bus to the DeviceNet bus.

Serial data transfer - multiple pieces of information transmitted one piece at a time.

Signal - an electrical or electro-magnetic transmission that indicates or represents an occurrence.

Signal generator - See *transmitter*.

Solicited messaging - a response to another node or a response when it is the node's predetermined time to speak (token passing).

Star - a term used to describe a bus topology where the datalines radiate from a single point.

Start of message - a certain number of high bits that start a clock so that a time reference is created to encode and decode the bits.

Stuff bit - an opposite polarity bit that is inserted in a string of consecutive polarity bits by the transmitter and removed by the receiver.

Tee - a product that creates a single branch or drop from a bus.

Terminating resistor - a resistor that is usually put at the beginning and end of a bus to stabilize and tune the signal.

Thickline - with respect to DeviceNet, this is an alternate name given to the *trunkline* cable.

Thinline - with respect to DeviceNet, this is an alternate name given to the *dropline cable*.

Token passing media access - this is a message shift method that is incremented in a manner that allows each node a chance to talk during each cycle.

Topology - See *bus topology*.

Transceiver - an electronic component that can transmit or receive a signal.

Transmitter - an electronic component that is coupled to the physical media. It is vocal chords of the central processing unit or other logic components.

Trunkline - the main busline.

Unsolicited messaging - an announcement of a change of condition.

Word - 2 bytes.